BAB II

TINJAUAN PUSTAKA

2.1 Human Computer Interaction

Human Computer Interaction adalah penelitian tentang interaksi antar manusi (*users*) dengan komputer. HCI sering dianggap sebagai bagian dari *computer science, behavioral science design* dan beberapa bidang pengetahuan lainnya. Interaksi antara *user* dan komputer terdapat pada *user interface* atau sering disebut *interface*, termasuk di dalamnya adalah *software* dan *hardware*.

Definisi HCI didefinisikan oleh Association for Computer Machinery adalah "Suatu disiplin ilmu yang berhubungan dengan desain, evaluasi dan implementasi dari sistem komputasi interaktif untuk kegunaan manusia dan pembelajaran mengenai fenomena-fenomena umum disekitarnya".

Karena HCI mempelajari keterkaitan manusia dan mesin, dibutuhkan pengetahuan dari kedua belah sisi mesin dan sisi manusia. Sisi mesin mempelajari teknik-teknik dalam *computer graphic*, *operating systems*, *programming language* dan *development environment* yang bersangkutan. Sedangkan dari sisi manusianya mempelajari tentang teori komunikasi, grafis dan desain industri, *linguistic*, ilmu sosial, *cognitive psychology*, dan *human performance*.

(Sumber: Alfa Faridh, Suni - Metafora Deteksi Tiupan Pada Game Flipscape, 2009)

2.2 Adobe Flex

Adobe Flex adalah sebuah platform pengembang aplikasi yang dapat digunakan untuk membangun dan memelihara aplikasi web ekspresif yang menyebar secara konsisten di semua browser utama, desktop, dan sistem operasi. *(Sumber: Wikipedia – Adobe Flex, 2008).*

Flex adalah aplikasi berbasis web namun aplikasi ini memberikan tingkat interaktivitas yang mendalam dengan media yang kaya pengalaman yang membuat tampilan lebih seperti program desktop komputer daripada aplikasi web tradisional. Flex merupakan *free open source framework* yang sangat produktif dengan teknologi yang ideal untuk digunakan ketika ingin membuat visualisasi data yang kompleks, kerangka kinerja, pengalaman multimedia, dan banyak aplikasi lainnya interaktif. Bahasa berbasis standar dan model pemrograman yang mendukung pola desain umum. MXML, sebuah bahasa berbasis XML-deklaratif yang digunakan untuk menggambarkan tata letak UI *(User Interface)* dan perilaku. ActionScript merupakan sebuah bahasa pemrograman berorientasi objek yang kuat, digunakan untuk membuat logika klien.

Flex juga termasuk komponen yang kaya pustaka dengan lebih dari 100 pustaka dengan komponen UI yang diperluas untuk membuat *Rich Internet Applications* (RIAs). *(Sumber: Noble, Joshua and Anderson, Todd – Adobe Flex 3 Cookbook, 2008)*

Gambar 2.1 Logo Adobe Flex

2.2.1 Definisi Adobe Flex

Definisi Flex dapat membingungkan karena Flex termasuk kombinasi dari teknologi yang berbeda. Flex bukan merupakan produk perangkat lunak tunggal, melainkan mencakup empat berikut potongan utama:

a. Bahasa: ActionScript 3.0 dan MXML menggunakan kombinasi dari ActionScript, bahasa scripting, dan MXML, bahasa markup, untuk membuat aplikasi Flex. MXML hampir sama seperti HTML, jadi apabila memiliki pengalaman membuat HTML halaman Web, maka cukup mudah untuk mengetahui MXML.
b. Komponen kerangka: Flex SDK Flex SDK (juga dikenal sebagai kerangka Flex) adalah satu set antarmuka pengguna komponen, seperti list, tombol, dan grafik, yang digunakan untuk membangun aplikasi Flex. The Flex SDK (dengan pengecualian memetakan paket) adalah bebas dan open source.
c. Integrated Development Environment (IDE) Flex Builder: Flex Builder digunakan untuk mengedit kode, kompilasi aplikasi, debug kode, dan kinerja profil. Flex Builder merupakan pengembangan terpadu lingkungan (IDE) yang dijual oleh Adobe.
d. Cross-Browser Runtime: Flash Player menjalankan aplikasi Flex di browser Web dengan *plug-in* Flash Player. Aplikasi Flex juga dapat digunakan sebagai desktop *standalone* atau aplikasi dengan menggunakan Adobe Integrated Runtime (AIR).

2.2.2 Sejarah Adobe Flex

Rilis awal Maret 2004 oleh Macromedia termasuk Software Development Kit (SDK), IDE, dan integrasi aplikasi J2EE dikenal sebagai Flex Data Services. Sejak Adobe mengakuisisi Macromedia tahun 2005, rilis berikutnya Flex tidak lagi memerlukan izin untuk Flex Data Services, yang telah menjadi produk terpisah berganti merek sebagai *LiveCycle* Data Services.

Pada bulan Februari 2008, merilis Adobe Flex 3 SDK open source di bawah Mozilla Public License. Adobe Flash Player, runtime yang aplikasi Flex dilihat, dan Adobe Flex Builder, IDE dibangun di atas platform open source Eclipse dan digunakan untuk membangun aplikasi Flex, tetap berpemilik.

Pada tanggal 26 April 2007 Adobe mengumumkan niat untuk melepaskan Flex 3 SDK (tak termasuk IDE Flex Builder dan LiveCycle Data Services) di bawah syarat-syarat Mozilla Public License. Adobe merilis versi beta pertama dari Flex 3, nama kode Moxie, pada bulan Juni 2007. Tambahan utama termasuk integrasi dengan versi baru dari Adobe Creative Suite produk, dukungan untuk AIR (Adobe aplikasi desktop baru runtime). Tambahan yang lain sebagai pendukung adalah ekstensi Cold Fusion dan penambahan profil dan refactoring alat untuk IDE Flex Builder.

a. Pada bulan Oktober 2007, Adobe merilis versi beta kedua dari Flex 3.
b. Pada tanggal 12 Desember 2007, Adobe merilis versi beta ketiga dari Flex
c. Pada tanggal 25 Februari 2008, Adobe Flex 3 dirilis dan Adobe AIR 1.0.
d. Pada tanggal 2 Juni 2010, Adobe Flex 4 dirilis.

2.2.3 Adobe Flex Framework

Flex Framework atau Kerangka Flex ini identik dengan *library class* Flex dan merupakan koleksi *class* ActionScript digunakan oleh aplikasi Flex. Flex Framework ditulis seluruhnya dalam *class* ActionScript, dan menetapkan kontrol, kontainer, dan manajer dirancang untuk memudahkan membangun *Rich Internet Applocations* (RIAs). Library kelas Flex adalah subyek banyak buku ini. Ini terdiri dari kategori berikut:

a. Form kontrol

   Form kontrol adalah kontrol standar seperti tombol, input teks, teks daerah, daftar, tombol radio, checkbox, dan combobox. Selain bentuk standar kontrol asing bagi pengembang HTML, pustaka kelas Flex juga mencakup kontrol seperti rich-text editor, pemilih warna, date selector, dan banyak lagi.

b. Menu kontrol

   Flex menyediakan satu set menu kontrol seperti pop-up menu dan menu bar.

c. Komponen Media

   Salah satu keunggulan dari aplikasi Flex adalah dukungan yang kaya media. Pustaka kelas Flex menyediakan seperangkat komponen untuk bekerja dengan media seperti gambar, audio, dan video.

d. Layout kontainer

   Flex memungkinkan aplikasi tata letak layar yang sangat dapat dikonfigurasi. Dapat menggunakan tata letak kontainer untuk

menempatkan isi dalam layar dan menentukan bagaimana akan berubah seiring waktu atau bila terdapat perubahan dimensi pada Flash Player. Dengan berbagai jenis komponen kontainer dapat membuat layout canggih dengan menggunakan grid, bentuk, kotak, kanvas, dan masih banyak lagi. Aplikasi tersebut juga dapat menempatkan elemen dengan absolut atau relatif koordinat sehingga dapat menyesuaikan dengan benar ke dimensi yang berbeda dalam Flash Player.

e. Data komponen dan data yang mengikat

Aplikasi Flex adalah aplikasi yang didistribusikan secara umum menjadikan prosedur remote panggilan ke layanan data yang berada di server. Komponen data terdiri dari konektor yang menyederhanakan prosedur panggilan, data model untuk menyimpan data yang akan dikembalikan, dan data fungsi mengikat secara otomatis data DNS bentuk asosiasi dengan model data.

f. Format dan validator

Data yang dikembalikan dari prosedur remote panggilan sering harus diformat sebelum mendapatkan ditampilkan kepada pengguna. Perpustakaan kelas Flex meliputi seperangkat fitur kuat dari format (format tanggal di berbagai representasi string, format nomor dengan presisi tertentu, format angka sebagai string nomor telepon, dll) untuk menyelesaikan tugas itu. Demikian juga, bila mengirim data ke layanan data dari input pengguna, perlu untuk dilakukan validasi data terlebih dahulu untuk memastikan sudah dalam bentuk yang benar.

g. Kursor manajemen

Tidak seperti aplikasi web tradisional, aplikasi Flex adalah aplikasi yang terkoordinasi dan tidak harus melakukan *refresh* layar menyelesaikan setiap data waktu dikirim atau diminta dari layanan data. Namun, karena prosedur remote panggilan sering dikenakan sistem jaringan dan latensi, penting untuk memberitahu pengguna ketika klien sedang menunggu di respon dari layanan data. Manajemen kursor Flex memungkinkan aplikasi untuk mengubah tampilan kursor untuk memberitahu pengguna perubahan tersebut.

h. State manajemen

Sebuah aplikasi Flex akan sering memerlukan perubahan banyak tempat. Sebagai contoh, operasi standar seperti mendaftar untuk account baru atau melakukan pembelian biasanya membutuhkan beberapa layar. *library class* Flex menyediakan kelas untuk mengelola perubahan itu dalam *state*. Manajemen tempat bekerja tidak hanya di tingkat makro untuk perubahan layar, tetapi juga di tingkat mikro untuk perubahan negara dalam komponen individu. Misalnya, komponen tampilan produk bisa beberapa negara: keadaan dasar hanya menampilkan gambar dan nama, dan rincian negara yang menambahkan deskripsi, harga, dan ketersediaan pengiriman. Selain itu, Flex menyediakan kemampuan untuk dengan mudah menerapkan transisi sehingga perubahan state animasi.

i. Efek

Aplikasi Flex tidak dibatasi oleh kendala aplikasi web tradisional. Sejak aplikasi Flex dijalankan dalam Flash Player, mereka bisa memanfaatkan fitur animasi Flash. Dengan demikian, perpustakaan kelas Flex memungkinkan beragam efek seperti memudar, membesarkan, mengaburkan, dan bersinar.

j. Sejarah manajemen

Sebagai tempat perubahan di dalam aplikasi Flex, fitur manajemen sejarah pustaka kelas Flex dimungkinkan untuk menavigasi dari negara bagian ke negara menggunakan bagian belakang dan tombol maju dari browser web.

k. Drag and drop manajemen

Pustaka kelas Flex menyederhanakan menambahkan fungsionalitas *drag and drop* untuk komponen dengan dibuat dalam satu bentuk, *drag and drop* pada komponen fungsionalitas pilih dan kelas manajer yang dimungkinkan dengan cepat menambahkan *drag and drop* pada komponen perilaku.

l. Alat tips

Gunakan fitur ini pustaka kelas Flex tips untuk menambah alat untuk unsur-unsur sebagai pengguna menggerakkan mouse ke atas. Alat tips berguna sebagai petunjuk maupun saran dalam menggunakan aplikasi tersebut.

m. Gaya manajemen

Pustaka kelas Flex memungkinkan untuk banyak mengontrol atas hampir setiap aspek dari aplikasi Flex. Seperti halnya aplikasi sejenis, aplikasi tersebut dapat melakukan perubahan gaya maupun tampilan seperti warna dan pengaturan font baik ukuran maupun jenisnya dengan kontrol yang mudah dari kontainer langsung kepada objek atau melalui css.

2.2.4 Kelebihan Adobe Flex Framework

Hal lain yang menarik terkait dengan pengembangan aplikasi Flex adalah:

a. Kemudahan dalam melakukan layouting aplikasi dengan fitur drag and drop komponen tersedianya panel properties dan event untuk masing-masing komponen, serta pengaturan posisi komponen dan tampilan aplikasi secara keseluruhan melalui fitur constraint.
b. Kemudahan koneksi ke sumber data dengan adanya fungsi terintegrasi untuk menyediakan RPC Service.
c. Konsep multilayer / page pada presentasi dengan menggunakann View States.
d. Kemudahan dalam membuat animasi transisi dan effect pada tiap-tiap komponen dan layer yang digunakan dalam pengembangan aplikasi yang lebih baik.
e. Compiler dan debugger yang terintegrasi dengan beberapa opsi pilihan output.

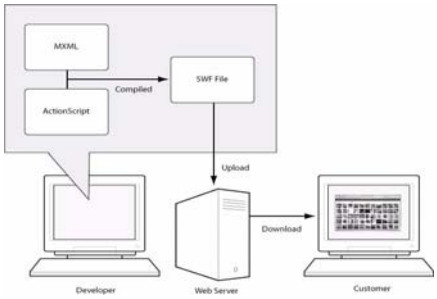
f. Cross platform yang dimiliki aplikasi web yang dapat benar-benar berubah menjadi aplikasi desktop.
g. Flex adalah teknologi masa depan dengan menunjukkan organisasi yang kompleks dan modern.
h. Flex bekerja pada semua platform utama dan pengguna tidak perlu menginstal apapun.
i. Audio dan video dapat diintegrasikan dengan baik, bahkan interaksinya lebih besar.

(Sumber: McCune, Dough and Subramaniam, Deepa – Adobe Flex 3.0 For Dummies, 2008)

2.2.5 Rich Internet Application (RIA)

Istilah Rich Internet Application (RIA) diciptakan Macromedia dan ditulis pada tahun 2002 untuk menggambarkan model baru dalam pengembangan aplikasi yang memisahkan back-end layanan data dari klien front-end kaya. RIA adalah aplikasi berbasis web yang dirancang untuk menyediakan fungsionalitas yang hampir sama dengan aplikasi desktop. Satu dari pilar pembangunan RIA adalah kemampuan untuk *asynchronously* (proses yang beroperasi secara independen dari proses lainnya) beban data di dalam aplikasi. Wikipedia HTML halaman Web membutuhkan satu halaman penuh refresh untuk memuat data baru. RIAs, di sisi lain beban data *asynchronous*, yang berarti mereka dapat memuat potongan data tanpa harus me-refresh halaman dan mereka melacak negara aplikasi dalam memori.

Beberapa RIAs dijalankan sepenuhnya di dalam web browser sementara yang lain browser-independen. RIA biasanya berjalan di dalam browser Web dan biasanya tidak memerlukan instalasi perangkat lunak di sisi klien untuk bekerja. RIAs menyediakan cara yang lebih kuat untuk berinteraksi dengan user dari pada aplikasi berbasis web tradisional. RIAs memungkinkan pengguna untuk melakukan penyuntingan pada baris, tarik dan taruh item dan dinyatakan berinteraksi langsung dengan unsur-unsur. RIAs juga memungkinkan hanya bagian halaman yang akan diperbarui, daripada harus *reload (mengisi kembali)* seluruh halaman. RIAs juga cenderung pada browser dan sistem operasi-independen. *(Sumber: Wikipedia – Rich Internet Application, 2008).*



Gambar 2.2 Alur Kerja Adobe Flex

2.3 Actionscript 3.0

ActionScript adalah bahasa pemrograman untuk Adobe ® Flash ® Player dan Adobe ® AIR ™ lingkungan run-time. Itu memungkinkan interaktivitas, penanganan data, dan banyak lagi di Flash, Flex, AIR dan konten dan aplikasi. ActionScript dijalankan oleh ActionScript Virtual Machine (AVM), yang merupakan bagian dari Flash Player dan AIR. ActionScript kode biasanya

dikompilasi ke dalam format bytecode (semacam bahasa pemrograman yang tertulis dan dipahami oleh komputer) oleh kompilator, seperti yang dibangun ke dalam Adobe ® Flash ® CS3 Professional atau Adobe ® Flex ™ Builder ™, atau yang tersedia dalam Adobe ® Flex ™ SDK dan Flex Data Services ™. Bytecode ini tertanam dalam file SWF, yang dijalankan oleh Flash Player dan AIR. ActionScript 3,0 menawarkan model pemrograman kuat yang akan menjadi familiar bagi pengembang dengan pengetahuan dasar tentang pemrograman berorientasi obyek. *(Sumber: Wikipedia – Actionscript 3.0, 2008).*

2.3.1 Kelas Dan Obyek

Dalam ActionScript 3.0, setiap objek didefinisikan oleh kelas. Kelas A dapat dianggap sebagai template atau cetak biru untuk mengetik objek. Definisi Kelas dapat mencakup variabel dan konstanta, yang memegang nilai-nilai data, dan metode, yang fungsi yang merangkum perilaku terikat kelas. Nilai-nilai yang disimpan dalam sifat dapat nilai primitif atau benda lainnya. nilai primitif adalah angka, string, atau nilai Boolean. ActionScript berisi sejumlah built-in kelas yang merupakan bagian dari bahasa inti. Beberapa dari built-in kelas, seperti Nomor, Boolean dan String, mewakili nilai-nilai primitif tersedia dalam ActionScript.Lainnya, seperti Array, Matematika, dan kelas XML, mendefinisikan objek yang lebih kompleks yang merupakan bagian dari standar ECMA-Script. Semua kelas, baik built-in atau user-defined, berasal dari kelas Objek. Untuk programmer Actionscript sebelumnya penting untuk dicatat bahwa tipe data obyek tidak lagi berupa default tipe data, meskipun

semua kelas - kelas lain masih berasal darinya. Dalam ActionScript 2.0, dua baris kode berikut adalah setara karena kekurangan dari penjelasan jenis berarti bahwa variabel akan tipe Obyek:

```
var someObj:Obyek;
var someObj;
```

ActionScript 3.0, bagaimanapun, memperkenalkan konsep variabel untyped, yang dapat ditunjuk dalam dua berikut cara:

```
var someObj: *;
var someObj;
```

Variabel *untyped* tidak sama sebagai variabel *someobject*. Perbedaan utama adalah bahwa variabel dapat untyped terus nilai khusus terdefinisi, sedangkan tipe variabel Obyek tidak bisa menahan nilai tersebut. Dapat didefinisikan sendiri dengan menggunakan kata kunci *class*. Pada pendeklarasian properti kelas dalam terdapat tiga macam cara: konstanta dapat didefinisikan dengan kata kunci *const,* variabel didefinisikan dengan kata kunci *var*, dan pengambil dan penyetel property didefinisikan dengan menggunakan *get* dan *set* atribut dalam sebuah deklarasi metode. Metode tersebut juga dapat dideklarasikan dengan fungsi kata kunci.

2.3.2 Paket Dan Namespaces

Paket dan namespaces adalah konsep terkait. Paket memungkinkan untuk pendefinisian kelas bersama-sama dengan cara yang memfasilitasi berbagi kode dan meminimalkan konflik penamaan. Namespaces

dimungkinkan untuk mengontrol visibilitas pengenal, seperti properti dan metode nama, dan dapat diterapkan untuk kode apakah itu berada di dalam atau di luar paket. Paket dimungkinkan untuk mengatur file kelas. Dan ruang nama dimungkinkan untuk mengatur penampakan sifat individu dan metode.

2.3.3 Variabel dan Konstanta

Sejak pemrograman terutama melibatkan perubahan potongan informasi dalam memori komputer, perlu ada suatu cara untuk mewakili potongan informasi dalam program. Variabel adalah nama yang mewakili nilai dalam memori komputer. Ketika menulis laporan untuk memanipulasi nilai, nama variabel ditulis di tempat nilai, dan komputer setiap saat akan melihat nama variabel dalam program serta menggunakan nilai yang ditemukan sana. Dalam ActionScript 3.0, variabel terdiri dari atas tiga bagian yang berbeda yaitu:

a. Nama variabel

b. Jenis data yang dapat disimpan dalam variabel.

c. Nilai aktual disimpan dalam memori computer

2.3.4 Tipe Data

Dalam ActionScript, terdapat banyak jenis data yang dapat digunakan sebagai tipe data dari variabel-variabel yang dibuat. Beberapa dapat dianggap sebagai "sederhana" atau "fundamental" tipe data:

a. String : nilai tekstual, seperti nama atau teks dari bab buku
b. numeric : ActionScript 3,0 mencakup tiga jenis data khusus untuk data numerik.
c. number : setiap nilai numerik, termasuk nilai-nilai dengan atau tanpa fraksi.
d. int : integer (bilangan bulat tanpa pecahan).
e. uint : sebuah "unsigned" integer, yang berarti seluruh nomor yang tidak boleh negatif.
f. Boolean : nilai benar atau salah, seperti apakah saklar adalah pada atau apakah dua nilai yang sama.

Sebagian besar built-in tipe data, serta jenis data yang didefinisikan oleh programmer, adalah jenis data yang kompleks. Beberapa jenis data yang kompleks yang mungkin dikenal adalah:

a. MovieClip : simbol klip video.
b. TextField : dinamis atau bidang input teks.
c. SimpleButton : lambang tombol.
d. Date : informasi tentang satu saat dalam waktu (tanggal dan waktu).

(Sumber: Lott,Joey and Schall,Darron – ActionScript 3.0 Cookbook, 2006; Tiwari,Shashank and Herrington,Jack – AdvancED Flex 3, 2008).

2.4 FLARToolKit

FLARToolKit adalah pustaka Actionscript 3.0 berdasarkan ARToolKit / NyARToolKit dimana ARToolKit berasal dari OpenCV menggunakan bahasa Visual C danNyARToolKit adalah versi java dari ARToolkit. FLARToolkit akan mendeteksi pola dari citra masukan *(marker detection)* dan menghitung posisi kamera dalam ruang dimensi tiga. FLARToolKit adalah apilkasi bebas (free or non-commerical application) di bawah lisensi GPL, yang berarti source code lengkap untuk aplikasi user.

Saat ini penggunaan FLARToolKit lebih diarahkan dalam pengembangan *marker detection* dengan produk yang dihasilkan adalah *Augmented Reality*. *(Sumber: Wikipedia – Adobe Flex, 2008).*

2.5 Marker Detection

Marker detection atau deteksi pola adalah deteksi pola merupakan deteksi sebuah pola pada suatu obyek dengan mengukur perubahan kecepatan, akurasi dan vektor dari suatu obyek yang terlihat pada tempat yang dituju. Pendeteksian pola ini dapat dicapai dengan alat mekanik yang dapat berinterksi dengan tempat atau obyek yang ukuran dan banyaknya perubahan dari lingkungan yang diteliti. *(Sumber: Wikipedia – Adobe Flex, 2008).*

Gambar 2.3 Contoh Marker (pola)

2.6 Augmented Reality

Ronald T. Azuma (1997) mendefinisikan *augmented reality* sebagai penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata, dan terdapat integrasi antarbenda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Penggabungan benda nyata dan maya dimungkinkan dengan teknologi tampilan yang sesuai, interaktivitas dimungkinkan melalui perangkat-perangkat input tertentu, dan integrasi yang baik memerlukan penjejakan yang efektif.

Selain menambahkan benda maya dalam lingkungan nyata, realitas tertambah juga berpotensi menghilangkan benda-benda yang sudah ada. Menambah sebuah lapisan gambar maya dimungkinkan untuk menghilangkan atau menyembunyikan lingkungan nyata dari pandangan pengguna. Misalnya, untuk menyembunyikan sebuah meja dalam lingkungan nyata, perlu digambarkan lapisan representasi tembok dan lantai kosong yang diletakkan di atas gambar meja nyata, sehingga menutupi meja nyata dari pandangan pengguna.

Penerapan dan Penggunaan Augmented Reality dapat dirasakan di berbagai bidang. Sebagai contoh dalam bidang Kesehatan adalah pada pemeriksaan sebelum operasi, seperti CT Scan atau MRI, yang memberikan gambaran kepada ahli bedah mengenai anatomi internal pasien. Dari gambar-gambar ini kemudian pembedahan direncanakan. Realitas tertambah dapat diaplikasikan sehingga tim bedah dapat melihat data CT Scan atau MRI pada pasien saat pembedahan berlangsung.

Bidang yang lain adalah bidang Militer. Kalangan militer telah bertahun-tahun menggunakan tampilan dalam kokpit yang menampilkan informasi kepada pilot pada kaca pelindung kokpit atau kaca depan helm penerbangan mereka. Dengan melengkapi anggota militer dengan tampilan kaca depan helm, aktivitas unit lain yang berpartisipasi dapat ditampilkan.

Bidang yang paling banyak menggunakan teknologi ini adalah bidang Hiburan. Bentuk sederhana dari *augmented reality* telah dipergunakan dalam bidang hiburan dan berita untuk waktu yang cukup lama. Contohnya adalah pada acara laporan cuaca dalam siaran televisi di mana wartawan ditampilkan berdiri di depan peta cuaca yang berubah. Dalam studio, wartawan tersebut sebenarnya berdiri di depan layar biru atau hijau. pencitraan yang asli digabungkan dengan peta buatan komputer menggunakan teknik yang bernama chroma-keying. Princeton Electronic Billboard telah mengembangkan sistem Augmented Reality yang memungkinkan lembaga penyiaran untuk memasukkan iklan ke dalam area tertentu gambar siaran. Contohnya, ketika menyiarkan sebuah pertandingan sepak bola, sistem ini dapat menempatkan sebuah iklan sehingga terlihat pada tembok luar stadium. *(Sumber: Wikipedia – Augmented Reality, 2008).*

Gambar 2.4 Contoh Augmented Reality

BAB III

ANALISA DAN PERANCANGAN

3.1. Analisa Aplikasi

Pada sub bab ini akan dibahas tentang perancangan aplikasi Driving Car Simulation menggunakan *marker detection* dengan *FlarToolKit dan Adobe Flex Framework* yang dikembangkan oleh penulis

Perancangan program ini dimaksudkan untuk mendapatkan suatu hasil yaitu berupa *software* aplikasi simulasi kendaraan dengan *marker detection*, sebagai sarana pembelajaran dan pengajaran berbasis flash. Teknik yang dilakukan yaitu dengan membuat aplikasi flash di web dengan memanfaatkan teknologi dari *Adobe Flex Framework* dengan menggunakan bahasa *Actionscript 3.0* yang nantinya akan digunakan pengguna (*user*) dengan memanfaatkan simulasi menggunakan *marker detection.*

Dengan memanfaatkan perangkat *webcam*, pengguna (*user*) dapat memainkan simulasi ini dimanapun dan kapanpun.

3.2. Perancangan Aplikasi

Perancangan aplikasi memberikan penjelasan mengenai cara kerja secara umum dan kebutuhan pengguna aplikasi. Rancangan alur program (*flowchart*) dimodelkan dengan *Microsoft Office Visio 2003*, sehingga tercipta tatap muka (*interface*) dari aplikasi.

3.2.1. Deskripsi Umum Aplikasi

Aplikasi simulasi ini berfungsi untuk menggerakkan atau mengontrol sebuah simulasi mobil menggunakan pola tertentu dengan menggunakan webcam. Untuk memainkan simulasi ini, user harus memiliki perangkat webcam untuk dapat menggerakkan karakter yang terdapat dalam simulasi ini.

Aplikasi simulasi dengan *marker detection* ini menggunakan perangkat *webcam* menggunakan *Adobe Flex Framework* dengan bahasa *Actionscript 3.0* memberikan peluang baru untuk mengembangkan berbagai aplikasi gerak.

3.2.2. Kebutuhan Aplikasi

Sistem yang dibangun berupa simulasi melalui media *webcam* serta menciptakan simulasi yang dapat dimainkan dengan mudah menggunakan *marker detection* berdasarkan gerakan pemainnya. Dengan demikian kebutuhan sistem dapat dikategorikan menjadi kebutuhan pengguna (*user*).

3.2.2.1. Kebutuhan Pengguna (*User*)

Aplikasi *driving car simulation* menggunakan *marker detection* dengan menggunakan teknologi *Adobe Flex Framework* dan bahasa pemrograman *Actionscript 3.0* adalah untuk melatih ketangkasan, kecepatan dan ketepatan dalam memainkannya.

Untuk memenuhi kebutuhan pengguna (*user*) mengenai interaksi dengan sistem dan untuk mengetahui kebutuhan-kebutuhan apa saja yang perlu dipersiapkan, maka perlu dijabarkan mengenai kebutuhan dari sudut

pandang perangkat *user*. Kebutuhan yang diperlukan oleh pengguna (*user*) adalah :

a. *Webcam* standar.

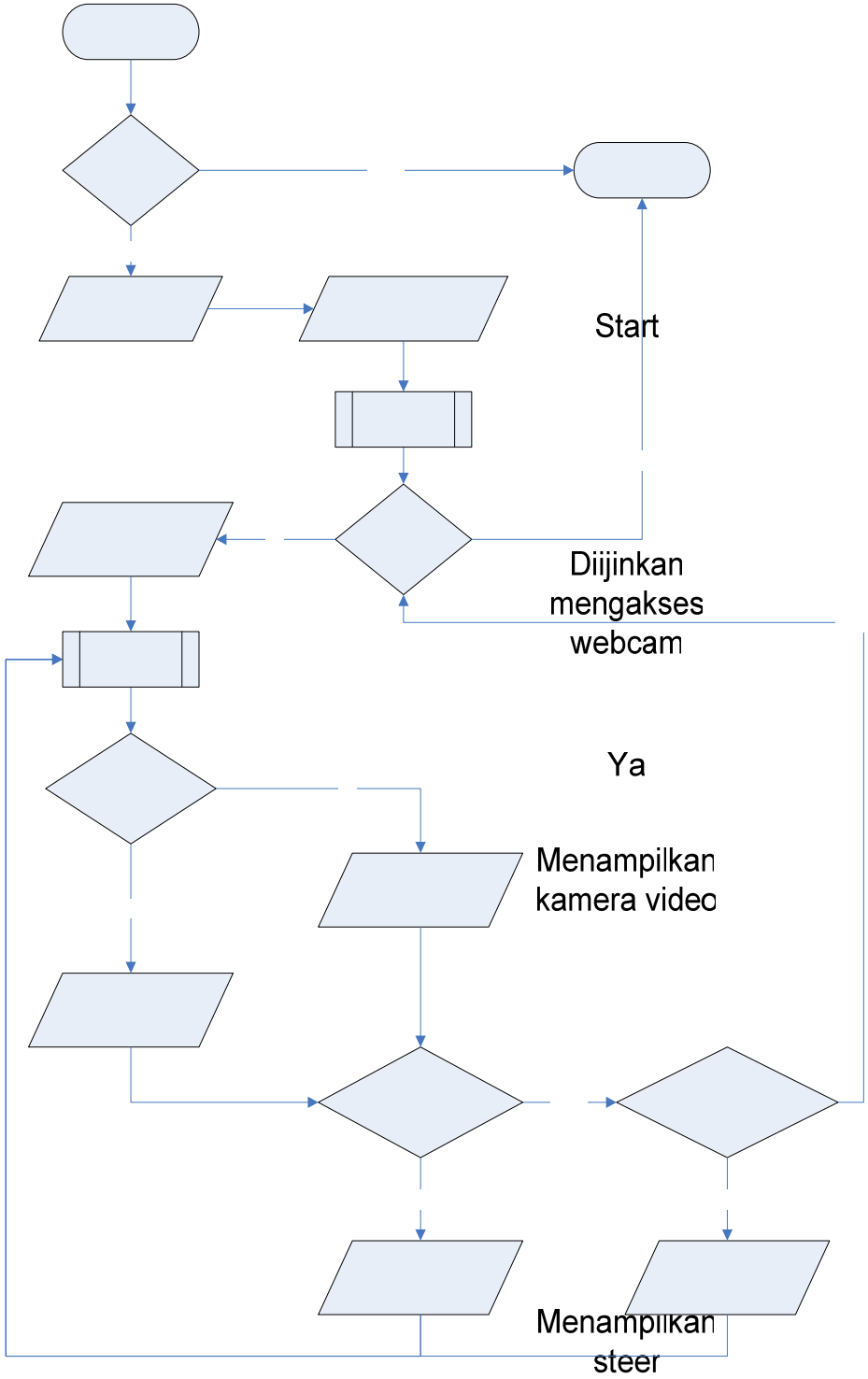
b. Aplikasi browser yang mendukung *Flash Player 10*

3.2.3. Flowchart

Flowchart adalah suatu gambaran yang menjelaskan tentang alir kerja aplikasi mulai dari awal sebelum aplikasi dijalankan hingga akhir dari aplikasi. Diambil dari (Sri,2002 pada www.total.or.id/info.php), *flowchart* merupakan bagan alir data yang menunjukkan sebuah alur data melalui program atau sistem penanganan informasi dan operasi-operasi yang dikenakan pada data proses-proses yang terpenting disepanjang jalur. Dengan kata lain *flowchart* berguna bagi sarana pembantu untuk menunjukkan bagaimana bekerjanya program yang diusulkan dan sebagai sarana untuk memahami operasi-operasi sebuah program. Dengan *flowchart* dapat digambarkan suatu tahapan penyelesaian program secara sederhana.

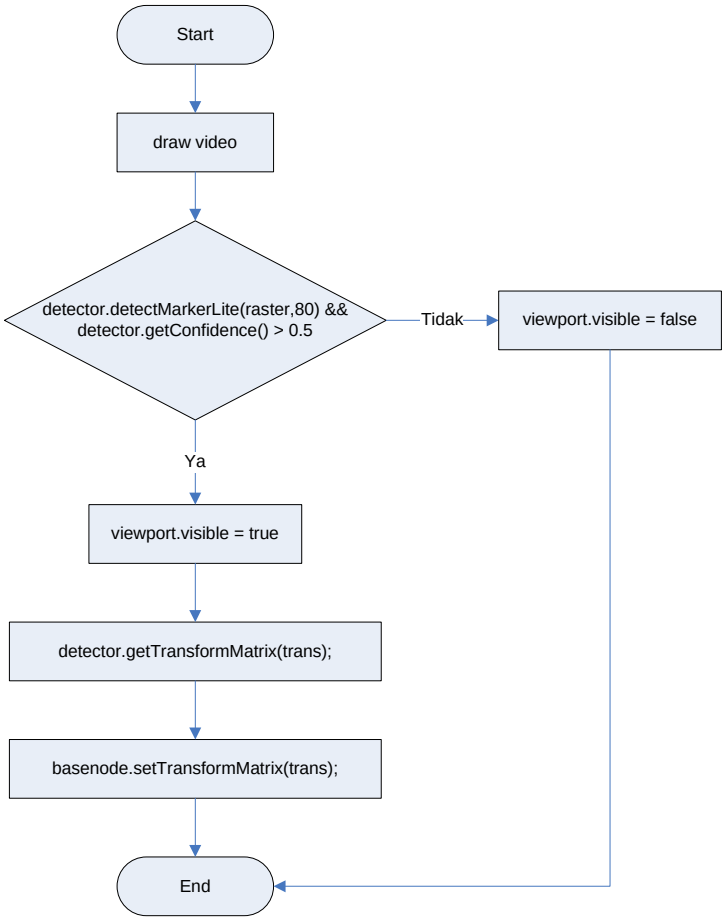
3.2.3.1. Flowchart Driving Car Simulation Menggunakan Marker Detection

Flowchart atau alur jalannya program simulasi mengemudi *(Driving Car Simulation)* dapat dilihat pada gambar 3.1 berikut.



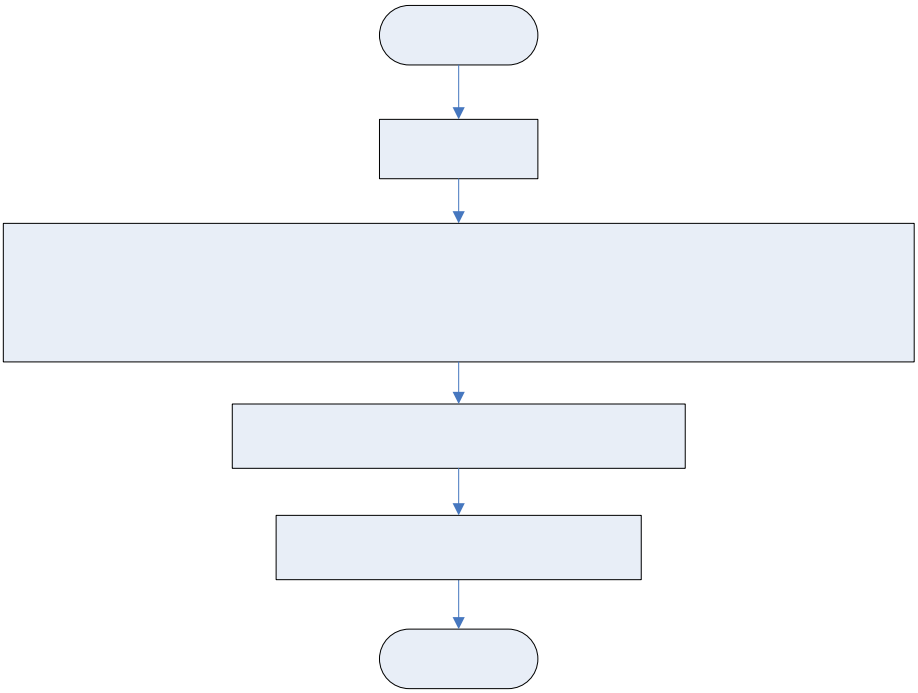
Gambar 3.1. Flowchart Driving Car Simulation

Gambar 3.1 menjelaskan alur pengguna (*user*) dalam menjalankan sistem. Pengguna diberikan pilihan untuk diijinkan mengakses webcam atau tidak. Jika tidak diijinkan maka program selesai namun jika diijinkan maka simulasi dapat dilanjutkan. Terdapat dua buah fungsi yang dipakai pada program tersebut yaitu fungsi deteksi marker dan fungsi kontrol stir. Pengguna dapat menggunakan *marker* yang sudah ditentukan sebelumnya. *Marker* tersebut di print untuk digunakan sebagai pengontrol mobil.



Gambar 3.2. Flowchart Fungsi Deteksi Marker

Flowchart Gambar 3.2 merupakan flowchart fungsi deteksi marker. Alur dari fungsi deteksi marker adalah ketika pengaksesan kamera diijinkan oleh user maka ditampilkan video. Kemudian dilakukan pengaksesan deteksi marker. Apabila deteksi FlarRgbRaster = raster dan threshold = 80 maka dan deteksi lebih dari 50%(0.5) maka ditampilkan viewport (viewport menamplkan animasi stir), pengambilan deteksi transform matrix, dan basenode transform matrix. Apabila tidak sesuai maka tidak ditampilkan viewport.



Gambar 3.3 Flowchart Fungsi Kontrol Stir

Flowchart Gambar 3.3 merupakan flowchart fungsi kontrol stir. Alur dari fungsi kontrol stir adalah ketika marker terdeteksi oleh kamera (webcam) maka dilakukan proses penghitungan matrix3D untuk

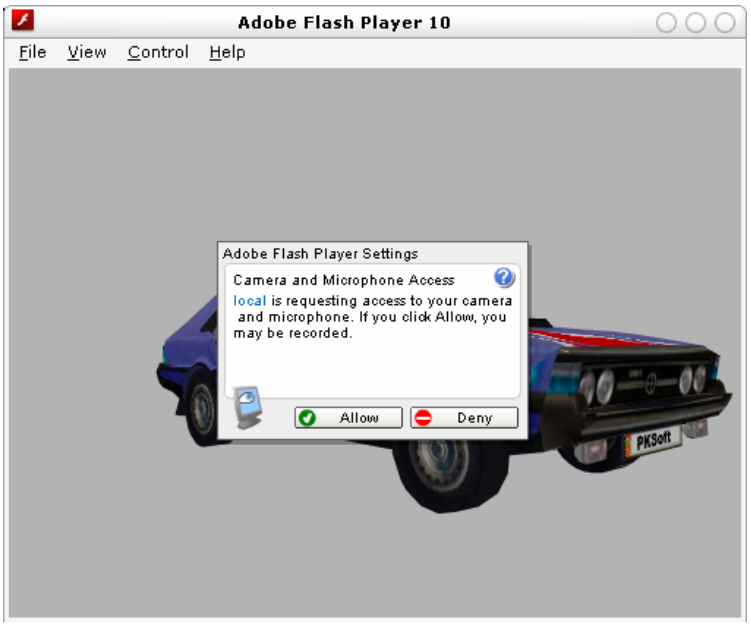
Ma

mendapatkan hasil posisi dan rotasi dari marker yang ditangkap oleh *webcam*. Posisi dan rotasi marker ini yang nantinya akan digunakan sebagai acuan dalam menggerakkan simulasi mobil.

3.2.4. Desain Antarmuka

Perancangan desain antarmuka dimaksudkan untuk memberikan gambaran bagaimana tampilan dari aplikasi *Driving Car Simulation* dengan *marker detection*. Adapun desain antar muka yang akan digunakan pada aplikasi ini adalah sebagai berikut :

a. Halaman konfirmasi kepada pengguna untuk diijinkan mengakses webcam atau tidak.



Gambar 3.4. Halaman Konfirmasi

b. Halaman Utama Simulasi.

Gambar 3.5. Halaman Utama

Gambar 3.6. Halaman Utama dengan Marker

Gambar 3.5 diatas adalah halaman utama simulasi. Pada halaman utama simulasi menampilkan kamera video (webcam) dan simulasi mobil tiga dimensi. Sedangkan pada Gambar 3.6 diatas terlihat *marker (pola)* yang terdeteksi pada webcam dan menampilkan sebuah *steer wheel (stir kemudi)* yang nantinya akan digunakan oleh user dalam mengontrol laju simulasi mobil tiga dimensi.

BAB IV

IMPLEMENTASI

Pada Bab IV ini akan dibahas mengenai implementasi dari rancangan sistem yang telah dibuat pada Bab III. Bagian implementasi sistem kali ini meliputi: lingkungan implementasi, implementasi proses, dan implementasi antarmuka.

4.1. Lingkungan Implementasi

Pada bagian implementasi akan dijelaskan mengenai perangkat lunak dan perangkat keras yang akan digunakan dalam pengimplementasiannya.

Perangkat Lunak :

a. Sistem operasi Microsoft Windows XP Professional Edition Service PackII
b. Java Virtual Machine: Sun JRE 1.4.2 atau yang lebih baru
c. Adobe® Flex™ Builder™ 3 Professional
d. Adobe® Flex™ 3 SDK
e. FLARToolKit Version 2.5.1
f. ARToolKit Marker Generator

Perangkat Keras :

a. Prosesor Intel® Pentium® 4, 2.8 GHz atau yang setara
b. DDR II Memori kapasitas 2 Giga Byte
c. Hard disk 5 Giga Byte

d. VGA 256 MB

e. Monitor

f. Keyboard

g. Mouse

h. Webcam

4.2. Penjelasan Program

Pada bagian ini akan dibahas tentang implementasi program yang merupakan hasil dari analisa dan perancangan sistem pada bab sebelumnya. Implementasi program ini ditujukan untuk operator dalam berinteraksi dengan sistem yang dihasilkan.

Sebelum memulai penjelasan harus melewati beberapa tahap untuk dapat menyelesaikan aplikasi ini. Adapun tahap-tahap yang harus dilalui antara lain :

a. Menginstall Adobe Flex SDK dan Adobe Flex Builder 3.0

b. Menambah library Flartoolkit ke dalam Adobe Flex

c. Membuat mobil tiga dimensi dan simulasi mobil.

d. Membuat *marker detection.*

e. Menggabungkan simulasi mobil dengan *marker detection.*

4.2.1. Menginstall Adobe Flex Builder 3.0

Saat mendownload Flex Builder 3 dari Adobe terdapat empat versi yang saat ini tersedia, dua untuk Windows dan dua untuk Mac. Untuk setiap sistem

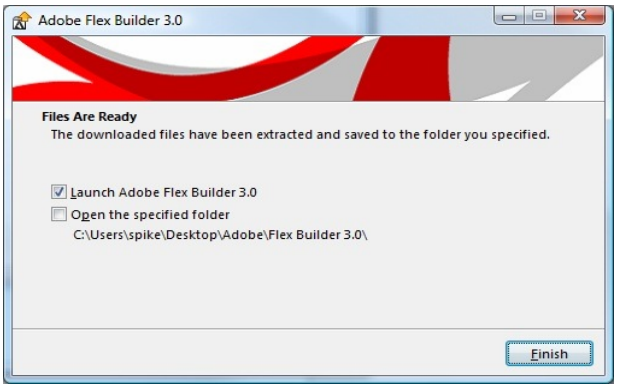
operasi ada dua versi, satu dikenal sebagai installer *standalone (berdiri sendiri)* dan lainnya sebagai sebuah *plug-in installer*.

Pada Flex Builder 3.0 sebenarnya terdapat dua installer terpisah. Apabila pengembang yang sudah menggunakan Eclipse sebelumnya untuk tujuan lain seperti pengembangan JAVA, mungkin memutuskan untuk menggunakan steker di installer. Apabila menginstal Flex Builder 3.0 pada versi terbaru, sudah termasuk instalasi Eclipse didalamnya. Hal ini memungkinkan untuk menyimpan semua *plug-in* yang ada dan kemudian menggunakan di Flex Builder 3.0 di atas lingkungan yang ada.

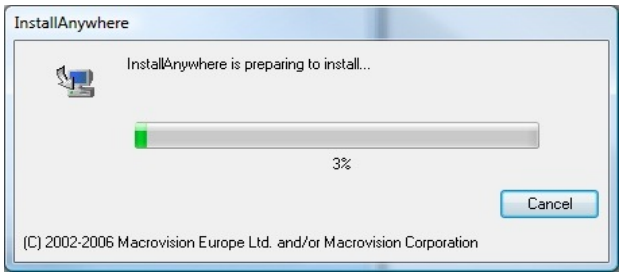
Jika baru menggunakan Eclipse disarankan untuk lebih baik menggunakan installer berdiri sendiri, yang menyertakan salinan Eclipse. Bahkan meskipun telah menggunakan installer mandiri, penting untuk mengetahui bahwa tidak hanya menginstal *plug-in* Flex Builder tetapi juga salinan lengkap dari Eclipse. Dan user bebas untuk menggunakan *plug-in* lain untuk platform lainnya. Misalnya, pengembang ColdFusion dapat menggunakan *plug-in* Eclipse bebas dalam pengembangan aplikasi ColdFusion. Bahkan jika installer mandiri, yang masih berjalan di Eclipse dasar, yang berarti mendapatkan *plug-in* tambahan dan meletakkannya di atas instalasi Flex Builder. User dapat mendownload di alamat http://www.adobe.com/products/flex/ kemudian lakukan installasi. Berikut ini adalah tahapan dalam melakukan installasi Adobe Flex Builder 3.0.

Apabila telah mendownload aplikasi Adobe Flex Builder 3.0, maka tahapan pertama yang dilakukan untuk intallasi adalah mengekstrak terlebih

dahulu seperti yang ditamilkan pada Gambar 4.1. Setelah selesai melakukan ekstraksi maka proses installasi dimulai dengan melakukan persiapan installasi seperti pada Gambar 4.2.



Gambar 4.1 Extract Adobe Flex Builder 3.0



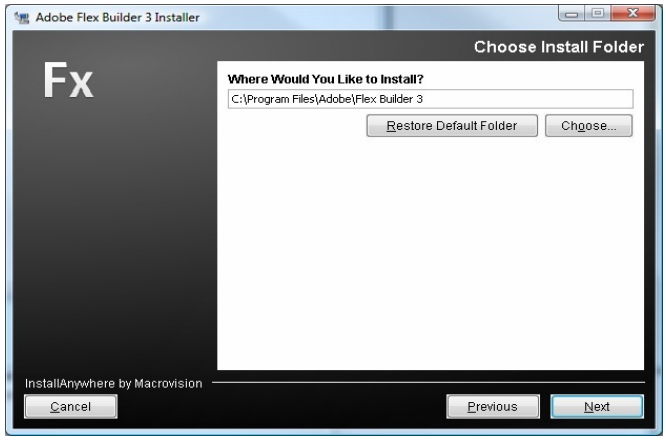
Gambar 4.2 Persiapan Installasi

Apabila selesai maka siap untuk melakukan instalasi. User dapat menginstal versi Windows yaitu Adobe Flex Builder 3.0 yang bekerja di Windows 2000/Xp/Vista. Hal tersebut penting untuk memastikan bahwa komputer memiliki cukup memori. Disarankan minimum 1GB RAM pada komputer. Dan apabila menjalankan basis data atau memori dengan tugas berat pada aplikasi lainnya, mungkin lebih baik untuk menambah memori menjadi 2 GB RAM atau lebih untuk mendapatkan hasil yang maksimal.

Ketika installer selesai melakukan persiapan,maka pertama kali yang disajikan pada layar informasi adalah menanyakan bahasa yang ingin digunakan untuk proses instalasi seperti yang tampak pada Gambar 4.3.

Gambar 4.3 Pilih Bahasa

Akan lebih baik jika membiarkan pada pilihan awal ke bahasa Inggris kemudian klik OK. Setelah beberapa saat, akan terlihat layar pengantar. Pastikan untuk menutup semua program sebelum melanjutkan dengan instalasi. Secara khusus, pastikan untuk menutup jendela browser apapun seperti Firefox atau Internet Explorer. Sebagai bagian dari instalasi harus menginstal ActiveX dan Plug dalam versi Flash Player. Dan perlu memastikan bahwa jendela browser benar-benar tertutup sebelum dapat melanjutkan proses tersebut. Klik tombol berikutnya, pada layar, jika menerima Persyaratan perjanjian lisensi. Klik tombol "I accept" tombol dan klik NEXT. Dan kemudian menunjukkan tempat di mana ingin menginstal Builder 3.0 Flex. Penulis menggunakan pengaturan default Program Files\Adobe\Flex Builder 3. Tetapi dapat menempatkan Flex 3 mana saja pada harddisk seperti tampak pada Gambar 4.4

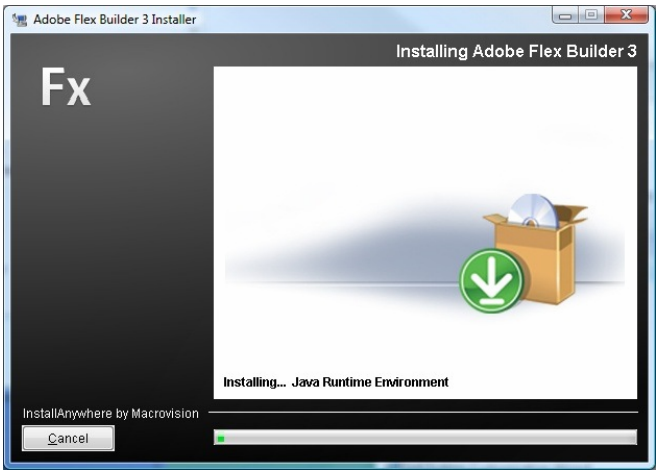


Gambar 4.4 Pilih Tempat Installasi

Pada layar berikut, akan ditanya alat apa yang ingin ditambahkan untuk menginstal. Secara umum program akan memilih flash Player sebagai syarat khusus penginstallan. Versi Flash Player di akan menginstal titik ini adalah debug Flash Player 9 dan Flash Player versi ini sangat penting untuk debugging dan dinyatakan pengujian aplikasi. Ada juga alat tambahan ekstensi ColdFusion tersedia disebut untuk FlexBuilder. ekstensi ColdFusion *plug-in* untuk Flex Builder yang dapat digunakan untuk pengembang ColdFusion. ekstensi ini meliputi kemampuan RDS *(Remote Data Services)*, Jasa Pembuatan Remote. Yang memungkinkan untuk menghubungkan server ColdFusion dan mendapatkan informasi struktur data dan menghasilkan instalasi berharga code.The JSEclipse JavaScript untuk pengembang. Plug-in JSEclipse adalah penggunaan khusus bagi pengembang untuk akan membangun dengan Adobe AIR atau aplikasi yang berkerja pada desktop. Dalam hal ini penulis menginstall semua pilihan yang ada seperti yang tampak pada Gambar 4.5.

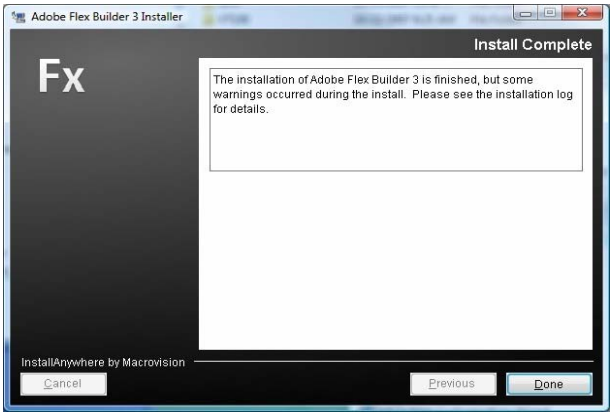
Gambar 4.5 Pilih Fitur Tambahan Adobe Flex

Setelah beberapa saat melihat ringkasan proses sebelumnya dari pilihan instalasi kemudian klik tombol Install untuk menyelesaikan instalasi. Proses instalasi membutuhkan waktu beberapa menit untuk berjalan tetapi setelah terinstal, maka dapat memulai Flex Builder 3 dan menjalankan aplikasi. Proses installasi seperti yang tampak pada Gambar 4.6.



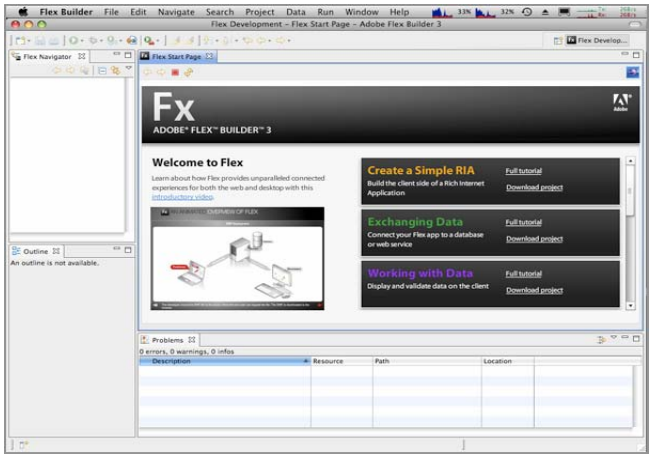
Gambar 4.6 Proses Installasi

Setelah proses instalasi selesai, maka layar akan menampilkan pesan bahwa aplikasi Adobe Flex Builder 3.0 telah selesai diinstal seperti yang tampak pada Gambar 4.7. dan siap untuk memulai membangun sebuah aplikasi.



Gambar 4.7 Selesai Installasi

Untuk menjalankan program silahkan mengakses program pada C:\Program Files\Adobe\Flex Builder 3\FlexBuilder.exe. Gambar 4.8 dibawah ini adalah tampilan awal dari Adobe Flex Builder 3.0.



Gambar 4.8 Tampilan Awal Adobe Flex Builder 3.0

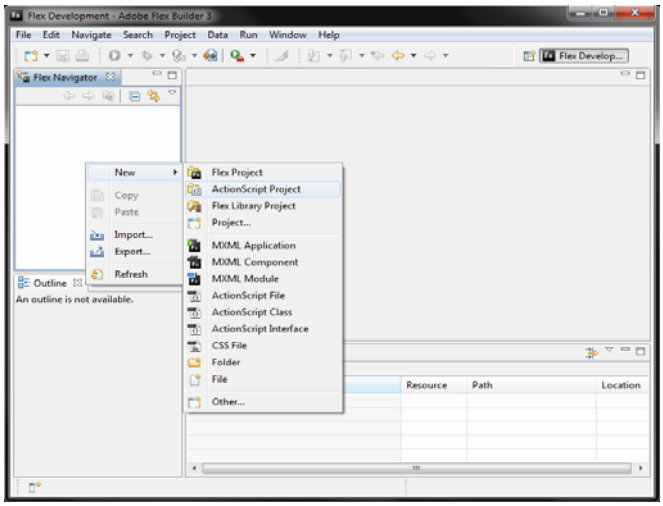
4.2.2. Membuat Proyek Actionscript Baru

Flex Builder dimungkinkan untuk membuat sebuah proyek ActionScript yang menggunakan API *(Application Proggramming Interface)* berbasis Flash (bukan kerangka Flex). Dengan memanfaatkan alat tersebut Flex Builder bekerja pada editor Actionscript, yang berarti bahwa memiliki IDE *(Integrated Development Environtment)* dengan fitur lengkap untuk mengembangkan aplikasi ActionScript.

Actionscript proyek tidak memiliki representasi visual di Flex Builder, dalam kata lain, tidak ada modus desain untuk aplikasi Actionscript. Untuk melihat aplikasi ActionScript tersebut hanya dengan cara mengkompilasi / menjalankannya pada Flex Builder dan kemudian berjalan dalam Flash Player.

Jika membuat sebuah proyek Actionscript baru, panduan wizard Proyek dapat melalui langkah-langkah yang telah ditentukan baik untuk menentukan jenis proyek, nama proyek, lokasi, dan opsi lanjutan lainnya.

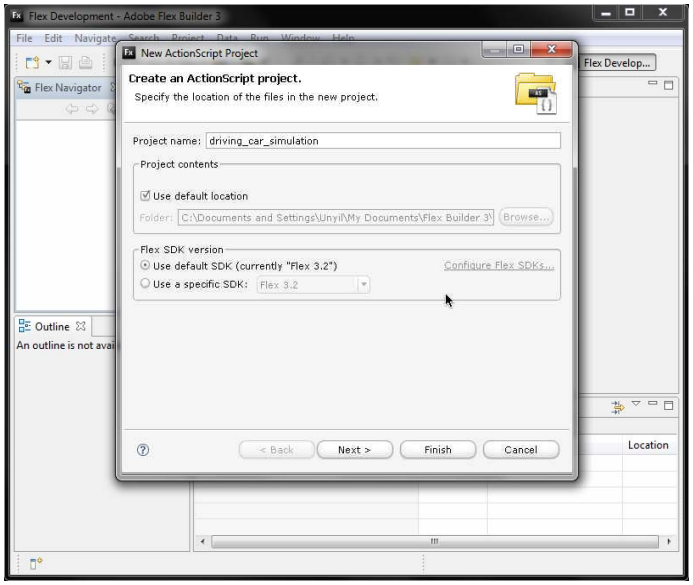
a. Pilih File > New > ActionScript Project.



Gambar 4.9 Membuat Proyek Actionscript Baru

b. Buat nama proyek baru

Tahap selanjutnya adalah dapat memilih sendiri lokasi / tempat menyimpan proyek. Dalam hal ini penulis memilih pengaturan default.



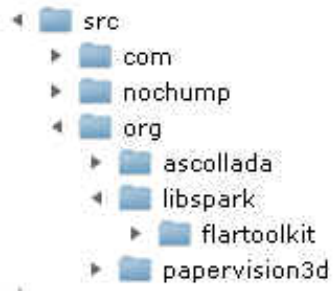
Gambar 4.10 Mengisi Nama Proyek Actionscript

c. Memilih folder sumber atau pustaka yang akan dipakai.

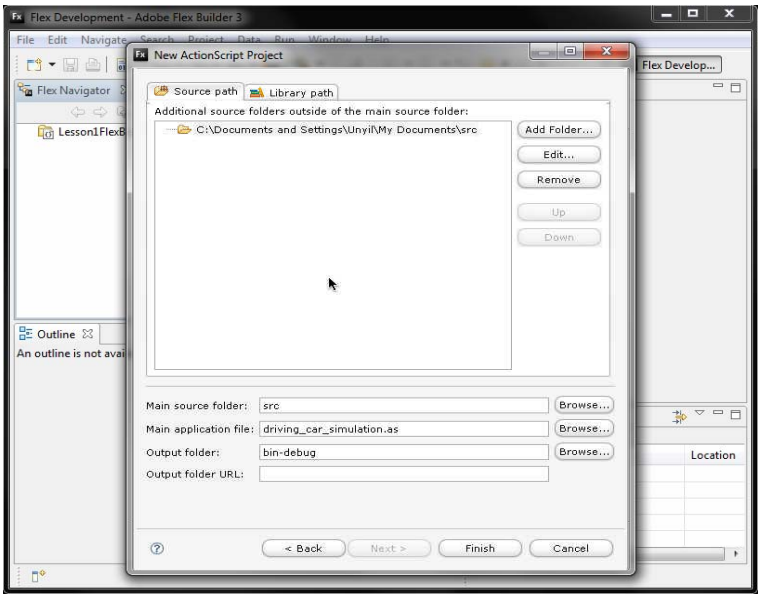
Bagian ini yang mengatur membangun jalan dan nama file aplikasi untuk proyek Actionscript. Mengatur folder sebagai sumber utama src file (singkatan dari "source" / sumber), dan aplikasi utama sebagai driving_car_simulation.as.

Sumber utama dan pustaka yang dipakai penulis adalah menggunakan Flartoolkit dan Papervision3D. Untuk Flartoolkit, dapat didownload di alamat http://www.libspark.org/wiki/saqoosha/FLARToolKit/en kemudian *extract* pada folder *src*. Untuk *library* / pustaka papervision3D, dapat didownload

dialamat http://code.google.com/p/papervision3d/downloads/list kemudian *extract* dan jadikan satu dengan folder *src*. Arahkan folder sumber ke folder *src* kemudian tekan Finish.



Gambar 4.11 Folder Sumber

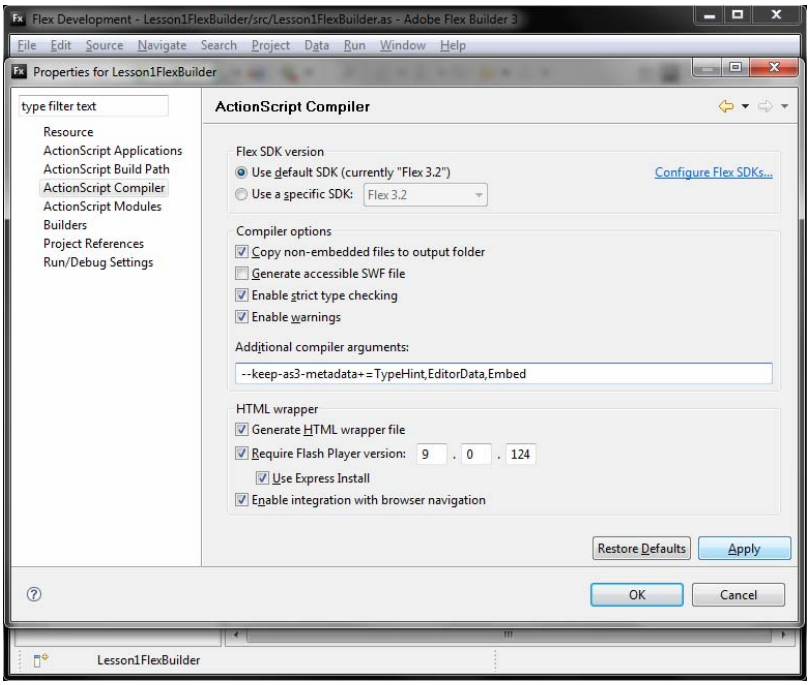


Gambar 4.12 Sumber dan Tempat Pustaka

d. Memilih *compiler* yang akan dipakai.

User dapat memilih *Actionscript Compiler* untuk mengatur apakah akan menampilkan project di web browser atau tidak. Pilih File yang dituju

klik kanan > Properties > ActionScript Compiler. Hilangkan Centang HTML wrapper apabila tidak ingin menampilkan proyek di web browser.



Gambar 4.13 Properties

4.2.3. Membuat Steering Wheel (Stir Kemudi) Augmented Reality Dan Marker Detector Dengan Adobe Flex Builder 3.0.

Pada bagian ini dijelaskan pembuatan program Augmented Reality dengan memanfaatkan marker detector untuk mendeteksi pola yang yang ditangkap oleh webcam. Pola tersebut akan digunakan sebagai pengontrol dari animasi mobil pada simulasi ini. Apabila pola tersebut ditangkap oleh webcam maka akan menampilkan animasi stir (steer) tiga dimensi.

Sebelum membuat program Augmented Reality terdapat beberapa kriteria yang dibutuhkan. Diantaranya adalah sebagai berikut:

a. Kamera parameter file
b. Marker File
c. Model 3D

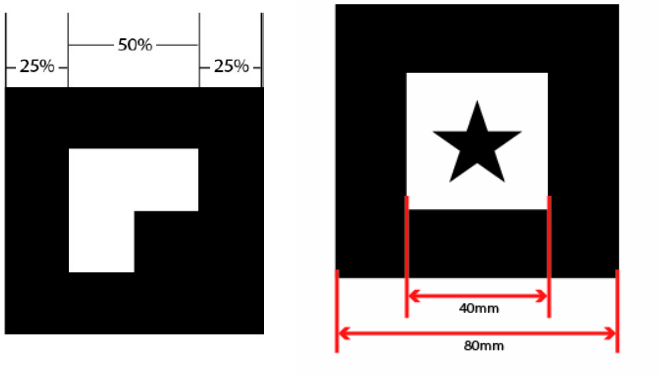
4.2.3.1. Kamera Parameter File

Kamera Parameter File / parameter kamera adalah file biner yang dimuat. File ini berasal dari program yang datang dengan ARToolkit. (Yang FLARToolkit app asli asli porting dari) Program yang menghasilkan file-file ini disebut "calib_camera2".

Calib_camera2 menciptakan file biner dan data dari file biner digunakan untuk mengoreksi distorsi dari lensa webcam. File tersebut dapat didownload, namun ada alternafif lain dengan menggunakan camera_para.dat yang sama dengan FLARToolkit.

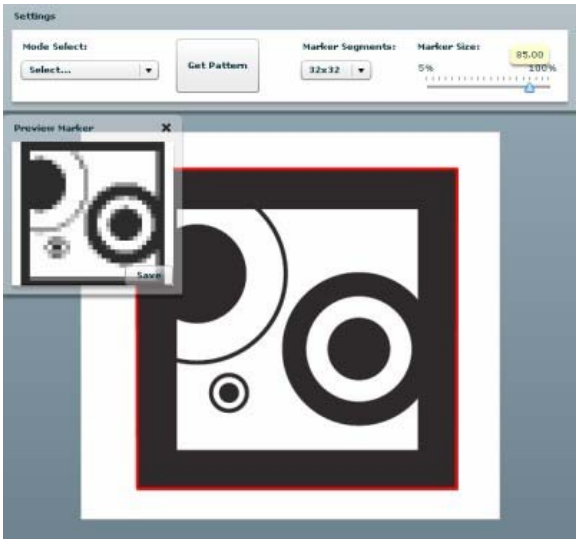
4.2.3.2. Marker File

Marker file adalah pola Flar akan dicari dan dideteksi dari webcam. Marker file yang dimaksud adalah file dalam bentuk “ *.pat". Pada marker diatas terlihat bahwa terdapat empat 16 × 48 matriks. Ini adalah *marker* yang dideteksi dari 4 arah yang berbeda. Flar akan mendeteksi *marker* sebagai barcode 16 × 16 2d. Dalam setiap file matriks adalah 16×48 karena sebenarnya 16 dengan 16 kali tiga warna (merah, hijau, dan biru). User dapat mendesain pola sendiri. Berikut ini adalah contoh pola yang baik untuk sistem.



Gambar 4.14 Marker (Pola)

ARToolkit Marker Generator Online digunakan untuk mendapatkan file dengan pola yang diinginkan seperti yang tampak pada Gambar 4.14. ARToolkit Marker Generator digunakan untuk menyimpan penanda sebagai pola file. FLARToolkit mendeteksi dan menggunakan untuk meningkatkan model3D. Berikut ini adalah alamatnya. http://flash.tarotaro.org/blog/2008/12/14/artoolkit-marker-generator-online-released/comment-page-1/



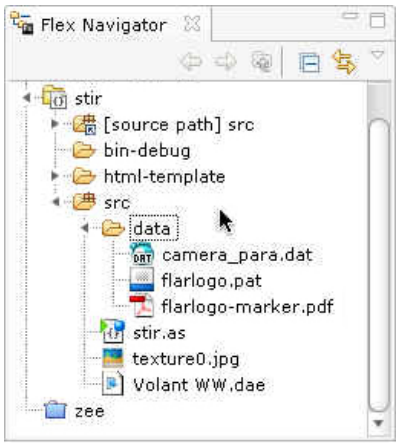
Gambar 4.15 ARToolkit Marker Generator Online

4.2.3.3. Model 3D

Untuk model tiga dimensi dapat dibuat sendiri sesuai yang diinginkan. Untuk membuatnya dapat digunakan alat permodelan seperti 3DMax, Blender, Cinema3D, dan lain sebagainya. Ubah model tiga dimensi menjadi collada atau " *.dae ". Atau dapat mendownload model tiga dimensi gratis di http://sketchup.google.com/3dwarehouse/ atau yang bebayar http://www.turbosquid.com/.

4.2.3.4. Membuat Augmented Reality Dan Marker Detection

Apabila kriteria diatas sudah dipenuhi maka pembuatan program dapat dimulai. Program yang dibuat menggunakan Adobe Flex Framework. Mulai dengan membuat proyek actionscript baru seperti yang sudah dijelaskan sebelumnya. Tambahkan kamera parameter file (camera_para.dat), marker file (flarlogo.pat), dan model tiga dimensi seperti (texture0.jpg dan Volant WW.dae) seperti yang tampak pada Gambar 4.16.



Gambar 4.16 Flex Navigator Stir

Berikut ini adalah source code yang digunakan dalam pembuatan program pendeteksian *marker*.

```
      [SWF(width="640",height="480", frameRate="30”)]

      public class stir extends Sprite
      {

        [Embed(source="/data/camera_para.dat",mimeType="applica
      tion/octet-stream")]
      private var params:Class;

        [Embed(source="/data/flarlogo.pat",
      mimeType="application/octet-stream")]
      private var pattern:Class;

      public function stir()
      {
            setupFLAR();
            setupCamera();
            setupBitmap();
            setupPV3D();
            addEventListener(Event.ENTER_FRAME,loop);
      }

      private function setupFLAR():void
      {
            fparams = new FLARParam();
            fparams.loadARParam(new params as ByteArray);
            mpattern = new FLARCode(16, 16);
            mpattern.loadARPatt(new pattern());
      }

      private function setupCamera():void
      {
            vid = new Video(320,240);
            cam = Camera.getCamera();
            cam.setMode(640,480,30);
            vid.attachCamera(cam);

            vid.x = 200;
            vid.scaleX *= -1;

            addChild(vid);
      }

      private function setupBitmap():void
      {
            bmd = new BitmapData(640,480);
            raster = new FLARRgbRaster_BitmapData(bmd);
            detector = new
FLARSingleMarkerDetector(fparams,mpattern,80);
      }
```

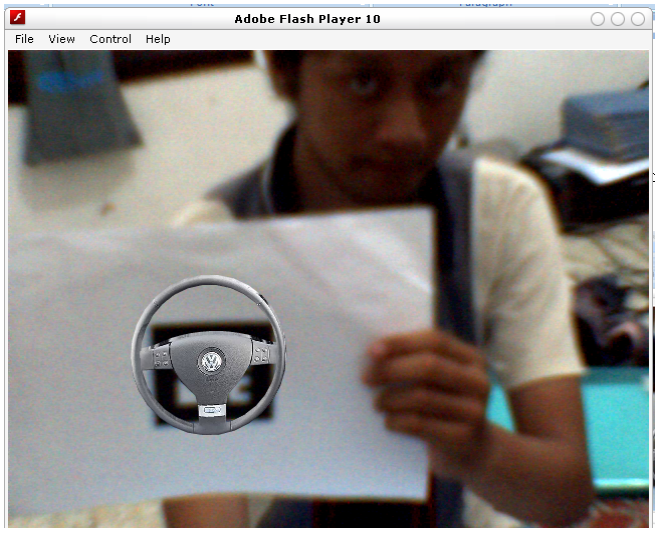
```
      private function setupPV3D():void
      {
            scene = new Scene3D();
            camera = new Camera3D;
            basenode = new FLARBaseNode();
            bre = new BasicRenderEngine();
            trans = new FLARTransMatResult();
            vp = new Viewport3D();

            var bitmap:BitmapFileMaterial;
            bitmap = new BitmapFileMaterial("texture0.jpg");

            var mil:MaterialsList = new MaterialsList({all:
bitmap});
            steer = new DAE();
            steer.load("Volant WW.dae",mil);
            steer.scale = 700;
            scene.addChild(basenode);
            scene.addChild(steer);
            addChild(vp);
      }

      private function loop(e:Event):void
      {
            bmd.draw(vid);
            try
            {
                  if(detector.detectMarkerLite(raster,80) &&
detector.getConfidence() > 0.5)
                  {
                        vp.visible = true;
                        detector.getTransformMatrix(trans);
                        basenode.setTransformMatrix(trans);
                  }
                  else
                  {
                        vp.visible = false;
                  }
            }
            catch(e:Error){}

        }
      }
```

Source code di atas digunakan dalam pembuatan pendeteksian program *marker*. Pustaka yang digunakan dalam pendeteksian adalah menggunakan FLARToolKit dengan menerapkan teknologi *augmented reality* sehingga dapat menampilkan sebuah animasi stir tiga dimensi.

Berikut ini adalah tampilan ketika source code diatas dijalankan.

Gambar 4.17 Pengaksesan Kamera

Gambar 4.17 diatas merupakan keluaran program dimana dilakukan pengijinan terhadap pengaksesan *webcam.*

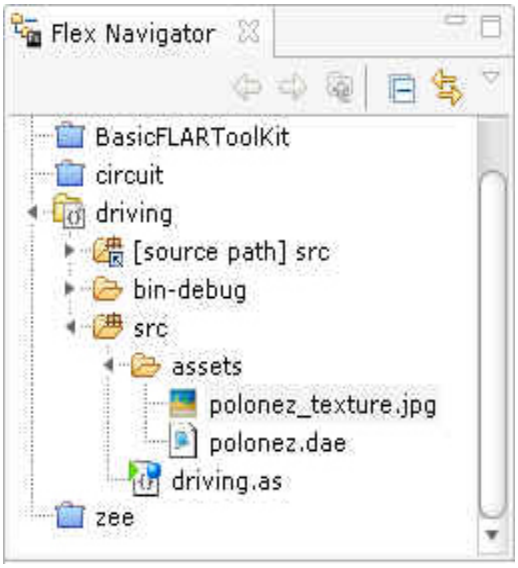


Gambar 4.18 Keluaran Program Menampilkan Stir

4.2.4. Membuat Mobil Tiga Dimensi Beserta Simulasinya Menggunakan Adobe Flex Builder 3.0.

Pada bagian ini akan dibahas mengenai pembuatan program simulasi mengemudi menggunakan Adobe Flex Builder 3.0. Simulasi yang penulis maksud adalah pembuatan simulasi mobil tiga dimensi dengan kontrol pada keyboard. Kontrol yang diinginkan adalah user dapat menggerakkan mobil tiga dimensi menggunakan keyboard. Adapun model mobil tiga dimensi yang digunakan adalah menggunakan model dalam bentuk file (*.dae) dapat diperoleh di http://sketchup.google.com/3dwarehouse/.

Pembuatan program dimulai dengan membuat proyek actionscript baru seperti yang sudah dijelaskan sebelumnya. Tambahkan model tiga dimensi seperti (texture0.jpg dan Volant WW.dae) seperti yang tampak pada Gambar 4.19.



Gambar 4.19 Flex Navigator Simulasi

Berikut ini adalah source code yang diperlukan.

```
	[SWF(frameRate="30", pageTitle="simulasi mobil",
backgroundColor="#FFFFFF")]
	public class driving extends BasicView
	{
	[Embed(source="/assets/polonez_texture.jpg")]
	public var carTextureAsset:Class;

	[Embed(source="/assets/polonez.dae",
	mimeType="application/octet-stream")]
	public var carModelAsset:Class;

	public function driving():void
	{
		if (stage) init();
		else addEventListener(Event.ADDED_TO_STAGE, init);
	}

	private function init(e:Event = null):void {
	removeEventListener(Event.ADDED_TO_STAGE, init);

	stage.quality = "high";

	initCameras();
	initTextures();
	initModel();

	statsClip = addChild(new StatsView(renderer)) as Sprite;
			statsClip.visible = false;

	handleResize();

	stage.addEventListener(Event.RESIZE, handleResize);

}
private function handleResize(e:Event = null):void {

	statsClip.y = stage.stageHeight - statsClip.height;
}
private function initCameras():void
{
	cameras[CAMERA_DEFAULT] = new Camera3D();
	cameras[CAMERA_DEFAULT].y = 5;
	cameras[CAMERA_DEFAULT].z = - 60;
	cameras[CAMERA_DEFAULT].zoom = 50;

	cameras[CAMERA_CAR_HOOD] = new Camera3D();
	cameras[CAMERA_CAR_BEHIND] = new Camera3D();
	cameras[CAMERA_CAR_WHEEL] = new Camera3D(30);

	this._camera = cameras[CAMERA_DEFAULT];
}
```

```
private function initModel():void
{
	carModelParent = new DisplayObject3D();
	scene.addChild(carModelParent);

	var mats:MaterialsList = new MaterialsList();
	mats.addMaterial(carModelTexture, "all");

	carModel = new DAE();
	carModelParent.x = 50;
	carModelParent.y = 0;
	carModelParent.z = 1050;
	carModelParent.rotationY = -35;

carModel.addEventListener(FileLoadEvent.LOAD_COMPLETE, onLoad);
	carModel.load(XML(new carModelAsset()), mats);

	carModelParent.addChild(carModel);
}
private function initTextures():void
{
	carModelTexture = new BitmapMaterial((new carTextureAsset
as Bitmap).bitmapData);

	carModelTexture.smooth = true;
}
private function onLoad(e:Event):void {
	carModel.scale = 100;
	wheelFrontRight = carModel.getChildByName(
"Wheel_Front_Right", true);
	wheelFrontLeft = carModel.getChildByName(
"Wheel_Front_Left", true );
	wheelRearLeft = carModel.getChildByName(
"Wheel_Rear_Left", true );
	wheelRearRight = carModel.getChildByName(
"Wheel_Rear_Right", true );

	cameraTargets[CAMERA_CAR_HOOD] = carModel.getChildByName(
"Camera_Hood_Target", true );
	cameraTargets[CAMERA_DEFAULT] = carModelParent;
	cameraTargets[CAMERA_CAR_BEHIND] =
carModel.getChildByName( "Licence_Plate_Front", true );
	cameraTargets[CAMERA_CAR_WHEEL] =
carModel.getChildByName( "Camera_Wheel_Target", true );

	enableListeners();
}
private function enableListeners():void{
	stage.addEventListener( KeyboardEvent.KEY_DOWN,
keyDownHandler, false, 0, true );
	stage.addEventListener( KeyboardEvent.KEY_UP,
keyUpHandler, false, 0, true );
	addEventListener(Event.ENTER_FRAME, tick, false, 0,
true);
}
```

```
public function tick(e:Event):void {
	updateCamerasPositions();
	camera.lookAt(cameraTargets[currentCamera]);
	driveCar();
	updateCarState();
	singleRender();
}
private function updateCamerasPositions():void{

	(cameras[CAMERA_CAR_WHEEL] as
Camera3D).copyTransform(carModelParent);
	(cameras[CAMERA_CAR_WHEEL] as Camera3D).moveForward(-
carModel.scale*1.0);
	(cameras[CAMERA_CAR_WHEEL] as
Camera3D).moveLeft(carModel.scale*3.8);
	(cameras[CAMERA_CAR_BEHIND] as
Camera3D).copyTransform(carModelParent);
	(cameras[CAMERA_CAR_BEHIND] as
Camera3D).moveForward(carModel.scale*6.5);
	(cameras[CAMERA_CAR_BEHIND] as
Camera3D).moveUp(carModel.scale*6);
	(cameras[CAMERA_CAR_HOOD] as
Camera3D).copyTransform(carModelParent);
	(cameras[CAMERA_CAR_HOOD] as
Camera3D).moveBackward(carModel.scale*2.993);
	(cameras[CAMERA_CAR_HOOD] as
Camera3D).moveUp(carModel.scale*1.25);

}
private function driveCar():void {

	if ( keyForward ) {
		maxSpeed = MAX_SPEED;
	} else if ( keyReverse ) {
		maxSpeed = MIN_SPEED;
	} else {
		maxSpeed = 0;
	}
	var acceleration:Number = ( speed - maxSpeed ) * 0.1;
	speed -= acceleration;

	if ( keyRight ) {
		if ( maxWheelAngle < 30) {
			maxWheelAngle += 2.0;
		}
	} else if ( keyLeft ) {
		if ( maxWheelAngle > -30 ) {
			maxWheelAngle -= 2.0;
		}
	} else {
		maxWheelAngle -= maxWheelAngle * 0.0416;
	}

	wheelAngle -= ( wheelAngle - maxWheelAngle ) * 0.5;
}
```

```
private function updateCarState():void {

      var movingForward:Boolean = speed > 0;
      var wheelRoll:Number = (0.7) * speed *
(carModel.scale/100);

      wheelFrontRight.rotationY  -= wheelRoll;
      wheelFrontLeft.rotationY += wheelRoll;
      wheelRearLeft.rotationY -=  wheelRoll;
      wheelRearRight.rotationY -=  wheelRoll;

      wheelFrontRight.rotationZ = 90-wheelAngle;
      wheelFrontLeft.rotationZ = -90-wheelAngle;

      carModelParent.yaw((carModel.scale/100)*speed *
wheelAngle * (0.002 + 0.0004 * Number(movingForward)) );

      carModelParent.moveBackward( (carModel.scale/100)*speed *
(0.9 + 0.2 * Number(movingForward)) );
}
private function keyDownHandler( event :KeyboardEvent ):void {
      switch ( event.keyCode ) {

            case "S".charCodeAt(0):
            case "s".charCodeAt(0):
                  statsClip.visible = !statsClip.visible;
                  break;

            case "C".charCodeAt(0):
            case "c".charCodeAt(0):

                  currentCamera++;
                  currentCamera %= cameras.length;
                  this._camera = cameras[currentCamera] as
Camera3D;
                  break;
            case Keyboard.UP :
                  keyForward = true;
                  keyReverse = false;
                  break;

            case Keyboard.DOWN :
                  keyReverse = true;
                  keyForward = false;
                  break;
            case Keyboard.LEFT :
                  keyLeft = true;
                  keyRight = false;
                  break;
            case Keyboard.RIGHT :
                  keyRight = true;
                  keyLeft = false;
                  break;
      }
}
```

```
private function keyUpHandler( event :KeyboardEvent ):void {
                    switch ( event.keyCode ) {
                          case Keyboard.UP :
                                keyForward = false;
                                break;
                          case Keyboard.DOWN :
                                keyReverse = false;
                                break;
                          case Keyboard.LEFT :
                                keyLeft = false;
                                break;
                          case Keyboard.RIGHT :
                                keyRight = false;
                                break;
                    }
              }

       }

}
```

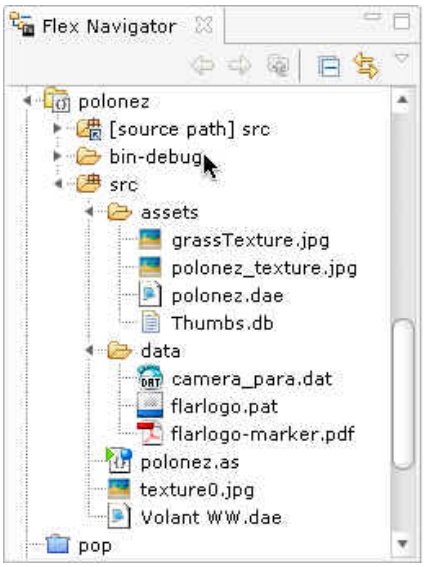
Apabila source code diatas dijalankan maka akan tampil keluaran seperti Gambar 4.20.

Gambar 4.20 Keluaran Program Simulasi

4.2.5. Menggabungkan Program Steering Wheel Augmented Reality Dengan Driving Car Simulation.

Pada bagian ini penulis melakukan percobaan dengan menggabungkan kedua buah program yang telah dibuat sebelumnya. Adapun program tersebut adalah program *steering wheel (stir kemudi)* Augmented Reality dan program *driving car simulation.* Hal ini dimaksudkan untuk mendapatkan kontrol mobil berdasarkan marker detection (deteksi pola) yang ditangkap oleh *webcam.*

Sama seperti program yang dibuat sebelumnya, dengan membuat proyek Actionscript baru kemudian memasukkan kedua buah *source code* tersebut kedalam sebuah proyek yang sama dimana *source code* tersebut akan dijalankan secara bersamaan. Hasil dari penggabungan tersebut tampak pada Gambar 4.21.



Gambar 4.21 Flex Navigator Polonez

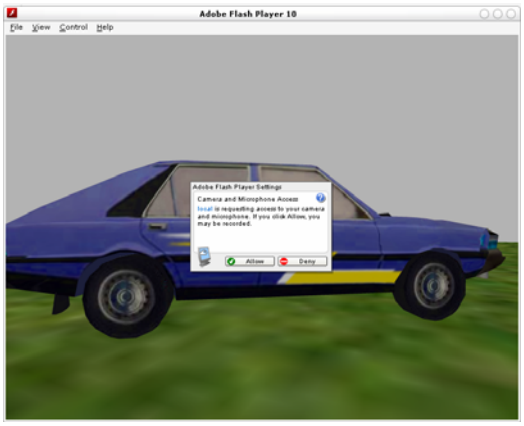
Berdasarkan program-program yang telah dikerjakan sebelumnya yaitu program *marker detection* dan *driving car simulation,* dilakukan

penggabungan antara keduanya untuk mendapatkan sebuah program baru. Adapun program baru tersebut dimaksudkan untuk membuat sebuah simulasi berkendara dengan kontrol deteksi pola *(driving car simulation based on marker detection).* Source code yang diperlukan sama dengan source code yang telah dibuat sebelumnya. Hanya diperlukan penggabungan dan ditambahkan source code berikut ini.

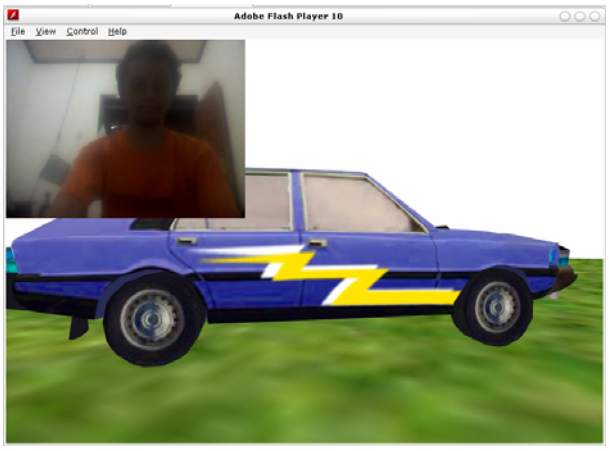
```
            var mat:Matrix3D = new Matrix3D();
            mat.n11 =  trans.m01; mat.n12 =  trans.m00; mat.n13
=  trans.m02; mat.n14 =  trans.m03;
            mat.n21 = -trans.m11; mat.n22 = -trans.m10; mat.n23
= -trans.m12; mat.n24 = -trans.m13;
            mat.n31 =  trans.m21; mat.n32 =  trans.m20; mat.n33
=  trans.m22; mat.n34 =  trans.m23;

            obPosition=new Number3D(mat.n14, mat.n24, mat.n34);
            obRotation=Matrix3D.matrix2euler(mat);

            if(obRotation)
            {
            var dAZ:Number=getShortestAngle(obRotation.z,
stir.rotationZ);
            var dAY:Number=getForwardBackward(obPosition.y,
stir.rotationY);
            var acceleration:Number = ( speed - maxSpeed ) *
0.1;
            speed -= acceleration;
            }
        }
        else{
            vp.visible = false;
        }
    }
    catch(e:Error){}
}

private function getShortestAngle(angle1:Number,
angle2:Number):Number
{
    var angle:Number=(angle1-angle2)%360;
    if(angle>1){
       maxWheelAngle -= 2.0;
    }
    else if(angle<-1){
       maxWheelAngle += 2.0;
    }
    return angle;
}
```

```
        private function getForwardBackward(forward:Number,
backward:Number):Number{
      var acel:Number = (forward-backward)%360;
      if ( acel > 10 ) {
            maxSpeed = MAX_SPEED;
      } else if ( acel < -10 ) {
            maxSpeed = MIN_SPEED;
      } else {
            maxSpeed = 0;
      }
      return acel;
}
```

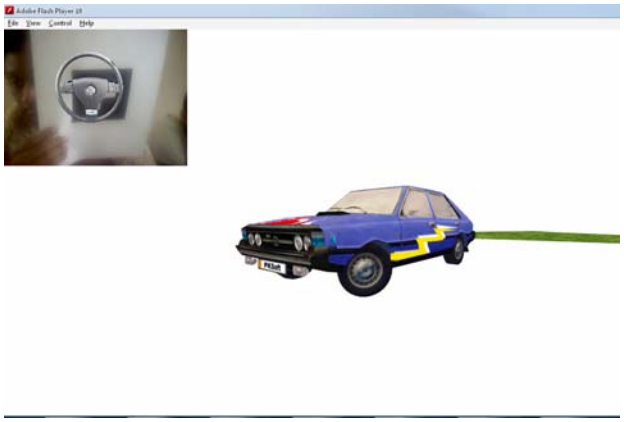
Pada source code tersebut ditambahkan sebuah matrix3D uantuk menghitung aktivitas pola *(marker)* saat terdeteksi. Aktivitas tersebut meliputi rotasi dan posisi pola *(marker)* yang dihitung saat terjadinya perubahan. Apabila terjadi perputaran pola (rotasi) maka dihitung sudut rotasi yang dibentuk. Bersarkan sudut tersebut maka roda akan berputar ke kanan atau ke kiri sejauh sudut yang dibentuk. Apabila terjadi perubahan posisi (translasi) maka dihitung jarak antara pola *(marker)* dengan webcam. Berdasarkan posisi tersebut maka mobil akan melaju ke depan atau kebelakang dengan kecepatan yang diatur sesuai dengan perubahan posisi. Berikut ini tampilan apabila source code diatas dijalankan.



Gambar 4.22 Keluaran Program Perijinan Mengakses Kamera



Gambar 4.23 Keluaran Program Dengan Menampilkan Simulasi Mobil



Gambar 4.24 Keluaran Program Simulasi Mobil Dengan Kontrol Pola Dari Webcam

Gambar 4.22 diatas merupakan keluaran program dengan perijinan untuk mengakses webcam user. Apabila diijinkan maka simulasi mobil dengan kontrol webcam dapat dilakukan. Apabila tidak diijinkan maka dapat menjalankan simulasi mobil menggunakan keyboard. Gambar 4.23 menampilkan simulasi mobil tiga dimensi. Gambar 4.24 menjalankan animasi mobil tiga dimensi menggunakan *marker detection.*

BAB V

UJI COBA DAN ANALISA

Pada bab ini akan dibahas mengenai uji coba terhadap aplikasi yang telah dibuat sebelumnya. Berdasarkan aplikasi tersebut selanjutnya akan dilakukan evaluasi sesuai dengan hasil uji coba tersebut. Uji coba dilaksanakan untuk mengetahui apakah aplikasi dapat berjalan dengan baik sesuai perancangan yang dibuat. Evaluasi dilakukan untuk menentukan tingkat keberhasilan dari aplikasi yang dibuat.

Uji coba yang akan dilakukan adalah uji coba pendeteksian pola dan uji coba simulasi mengemudi. Masing-masing uji coba disertai dengan tabel dan penjelasan mengenai uji coba yang dilakukan.

5.1. Uji Coba Pendeteksian Pola

Uji coba pendeteksian pola dilakukan untuk mengetahui apakah pola yang ditentukan sudah sesuai dan dapat dideteksi oleh program dengan baik dan benar. Berikut ini adalah uji coba yang dilakukan pada pendeteksian pola. Adapun uji coba pendeteksian pola *(marker)* dilakukan dengan 2 tahap, yaitu pendeteksian pola sejenis dan pendeteksian pola yang berbeda jenis. Hal ini dimaksudkan untuk menemukan pola *(marker)* yang terbaik yang dapat ditangkap webcam sehingga dapat menjalankan program dengan baik sesuai yang diharapkan.

5.1.1. Uji Coba Pendeteksian Pola Sejenis

Uji coba pendeteksian pola sejenis menggunakan pola yang telah diset sebelumnya pada program sebagai pola utama seperti yang tampak pada Gambar 5.1. Pola tersebut digunakan sebagai pola pembanding dan diujicobakan dengan pola *(marker)* yang sejenis namun memiliki kriteria yang berbeda baik pola, ukuran, intensitas ketebalan pola, dan lain sebagainya.

Gambar 5.1 Pola *(marker)* Utama

Berikut ini adalah tabel uji coba yang dilakuakan berdasarkan pola yang sejenis.

Tabel 5.1 Perbandingan Marker (Pola) Sejenis

No	Marker (Pola)	Marker Detection (Deteksi pola) Melalui webcam	Hasil	Status	Keterangan
1.			Terdeteksi	Kuat	Pola bawaan yang disediakan
2.			Tidak Terdeteksi	-	Beda pola

Lanjutan Tabel 5.1 Perbandingan Marker (Pola)

3.			Tidak Terdeteksi	-	Beda pola
4.			Terdeteksi	Lemah	Pola sama dengan tampilan mengkilat sebagian
5.			Terdeteksi	Lemah	Pola sama dengan ukuran diperkecil
6.			Tidak Terdeteksi	-	Pola sama dengan warna kebalikan

Keterangan:

Status ditentukan berdasarkan tingkat kemunculan stir pada pendeteksian pola oleh webcam dengan jarak yang sama dan dalam waktu 30 detik.

Kuat : kurang 3 kali

Lemah : lebih dari 3 kali

Berdasarkan tabel 5.1 diatas diperoleh hasil bahwa tidak semua *marker(pola)* yang user gunakan dapat dikenali oleh sistem. Hanya pola yang telah diset sebelumnya dan beberapa pola yang hampir sama yang dapat dikenali oleh sistem. Namun beberapa pola yang hampir sama menunjukkan intensitas pendeteksian yang lemah atau hanya sesekali dideteksi oleh sistem.

5.1.2. Uji Coba Pendeteksian Pola Berbeda Jenis

Uji coba yang dilakukan ini hampir sama dengan uji coba yang telah dilakukan sebelumnya. Pada uji coba ini dilakukan pembedaan pada pendeteksian pola. Pola yang digunakan adalah pola utuh dengan tidak ada ruang kosong pada bagian dalam pola. Adapun pola yang dideteksi sebagian menggunakan pola bangun datar yang telah diset sebelumnya pada program. Berikut ini adalah tabel uji coba yang telah dilakukan.

Tabel 5.2 Perbandingan Marker (Pola) Berbeda Jenis

No	Marker (Pola)	Marker Detection (Deteksi pola) Melalui webcam	Hasil	Status	Keterangan
1.			Terdeteksi	Kuat	Kontrol tidak sesuai
2.			Terdeteksi	Kuat	Kontrol tidak sesuai
3.			Terdeteksi	Kuat	Kontrol tidak sesuai
4.			Terdeteksi	Kuat	Kontrol tidak sesuai

Lanjutan Tabel 5.2 Perbandingan Marker (Pola) Berbeda Jenis

Keterangan:

Status ditentukan berdasarkan tingkat kemunculan stir pada pendeteksian pola oleh webcam dengan jarak yang sama dan dalam waktu 30 detik.

Kuat : kurang 3 kali

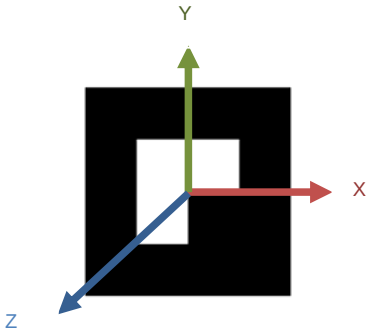
Lemah : lebih dari 3 kali

Berdasarkan tabel perbandingan diatas diperoleh hasil bahwa semua *marker (pola)* yang user gunakan dapat dikenali oleh sistem. Namun kontrol mobil tiga dimensi berdasarkan pola-pola tersebut tidak sesuai dengan pergerakan mobil maupun posisi dari animasi stir. Pada percobaan no.1. dan no.4 dengan pola *(marker)* persegi dan jajar genjang, menampilkan animasi stir dengan arah posisi stir tidak menentu demikian juga pergerakan mobil baik ke kanan, ke kiri ke depan maupun ke belakang. Pada percobaan no.2. dan no.3 dengan pola *(marker)* persegi panjang dan trapesium menampilkan animasi stir dengan arah posisi stir menghadap keatas atau proyeksi stir atas dan pergerakan mobil tidak menentu baik kedepan maupun kebelakang. Semua pola *(marker)* yang ditangkap oleh webcam dapat dideteksi, dapat menampilkan stir, dan menjalankan animasi mobil. Namun kontrol animasi mobil dengan pola-pola tersebut tidak sesuai dengan yang diharapkan. Oleh karena itu penggunaan pola-pola *(marker)* tersebut pada program tidak dianjurkan karena tidak sesuai dan tidak efisien untuk program.

5.2.Uji Coba Simulasi Mengemudi

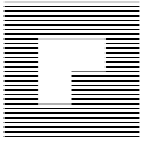
Uji coba simulasi mengemudi *(driving car simulation)* dilakukan untuk mengetahui apakah simulasi yang dibuat sudah sesuai yang diharapkan. Dalam uji coba kali ini dilakukan proses analisa terhadap laju pergerakan mobil berdasarkan kondisi *marker (pola)* yang ditangkap oleh *webcam.* Adapun *marker (pola)* yang dianalisa adalah meliputi translasi dan rotasi *marker* yang dilakukan oleh user.

Berikut ini adalah uji coba pergerakan mobil berdasarkan rotasi *marker* yang dilakukan oleh penulis.



Gambar 5.2 Sumbu X, Y, Z Marker

Tabel 5.3 Uji Simulasi Kendaraan (Driving Car Simulation)

No	Gambar Simulasi	Pola *(Marker)*	Sudut Rotasi	Translasi	Sumbu
1.			0 derajat	-	-

Lanjutan Tabel 5.3 Uji Simulasi Kendaraan (Driving Car Simulation)

2.			45 derajat	-	Sumbu Z
3.			- 45 derajat (315 derajat)	-	Sumbu Z
4.			-	Lebih dari 30 unit	Sumbu Y
5.			-	Kurang dari 30 unit	Sumbu Y

Keterangan:

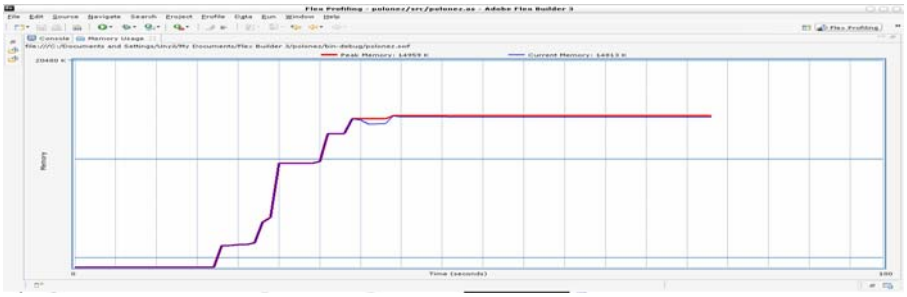
1 Unit = 1 cm dari jarak antara marker dengan webcam

Tabel 5.3 merupakan hasil uji coba terhadap aplikasi yang dibuat. Pada pada tabel no.1 simulasi dikerjakan menggunakan pola dengan sudut rotasi 0 derajat. Hasil yang dicapai adalah mobil berada dalam posisi diam atau roda dalam kondisi awal. Pada tabel no.2 marker kekanan sebesar 45 derajat diputar terhadap sumbu Z. Hasil yang dicapai adalah roda mobil bergerak kekanan. Pada tabel no.3 marker diputar terhadap sumbu Z atau kekiri sebesar -45 derajat terhadap sumbu Z. Hasil yang dicapai adalah roda mobil bergerak kekiri. Pada tabel no.4 marker ditranslasikan lebih dari 30 cm berdasarkan jarak antara webcam dengan marker. Hasil yang dicapai adalah roda mobil bergerak kedepan atau laju mobil kedepan. Pada tabel no.5 marker ditranslasikan kurang dari 30 cm berdasarkan jarak antara webcam dengan marker. Hasil yang dicapai adalah roda mobil bergerak kebelakang atau laju mobil kekebelakang.

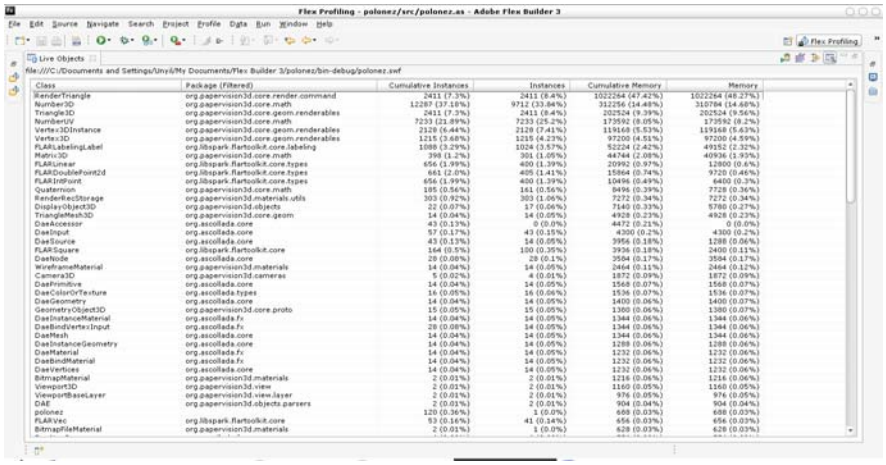
Berdasarkan uraian diatas diperoleh hasil bahwa uji simulasi yang dilakukan oleh penulis telah berhasil. Adapun kontrol mobil berdasarkan pergerakan *marker (pola) d*apat dilaksanakan dengan baik dan memperoleh hasil yang diinginkan.

Pada Adobe Flex Builder 3.0 terdapat komponen yang dapat membantu user dalam pembuktian dan mengidentifikasi hambatan kinerja maupun kebocoran memori di aplikasi yang dijalankan. Proses pembuktian pada Adobe Flex Builderdapat dikalukan ketika berinteraksi dengan aplikasi yang dijalankan. Fitur ini dapat membantu dalam membuktikan data tentang keadaan aplikasi, termasuk jumlah objek,

ukuran objek-objek, jumlah panggilan metode, dan waktu yang dihabiskan dalam panggilan-panggilan metode. Gambar 5.2 menunjukkan grafik pemakaian memori *(memory usage graph)* yang digunakan saat program dijalankan.



Gambar 5.3 Grafik Pemakaian Memori



Gambar 5.4 Live Objects

Pada gambar 5.3 diatas ditampilkan informasi-informasi mengenai kelas-kelas yang sedang jalankan oleh aplikasi tersebut. Pada gambar tersebut terlihat kelas *RenderTriangle* dengan *package: org.papervision3d.core.render.command* dengan akumulasi memori yang dibutuhkan sebesar 1022264 atau dalam prosentase sebesar 48,27%.

BAB VI

PENUTUP

6.1. Kesimpulan

Setelah melakukan uji coba dan evaluasi aplikasi, maka kesimpulan yang dapat diambil dari Simulasi mengemudi *(Driving Car Simulation)* yaitu:

a. Adobe Flex Framework merupakan aplikasi yang kompleks kerena berbasis OOP *(Object Oriented Programming)*. Walaupun demikian Adobe Flex mudah untuk dipelajari dan diaplikasikan meskipun belum umum digunakan di Indonesia.
b. Penerapan teknologi *marker detection* yang dilakukan telah berjalan dengan baik. Marker tersebut digunakan untuk mengontrol pergerakan maupun laju mobil pada simulasi kendaraan melalui media *webcam*.
c. Aplikasi yang dibuat merupakan inovasi awal dalam pengembangan sebuah model simulasi maupun permainan dengan menggunakan media *webcam*. Dengan bantuan *webcam,* user dapat berinteraksi dengan simulasi mobil tiga dimensi menggunakan *marker* layaknya menyetir sebuah mobil sungguhan. Aplikasi ini juga dapat dijalankan melalui web browser sehingga apabila ditanamkan dalam web server maka user dapat mengakses dimana saja dan kapan saja.
d. Aplikasi yang dibuat dengan metode pendeteksian pola *(marker detection)* dapat dikembangkan menjadi sebuah aplikasi yang nyata dan menarik *(Augmented Reality)*. Aplikasi ini juga tergolong murah karena tanpa

menggunakan sebuah *joystic* maupun *steering wheel* dapat menggerakkan sebuah animasi tiga dimensi. Aplikasi ini cukup menggunakan *webcam* dan Kertas berpola yang telah di cetak sebelumnya.

6.2. Saran

Beberapa saran yang dapat diberikan untuk pengembangan lebih lanjut. Adapun saran – saran tersebut adalah sebagai berikut:

a. Sebagai aplikasi *driving car simulation* perlu adanya peningkatan dalam perbaikan *interface* agar tampilan lebih menarik seperti perbaikan background dan animasi mobil yang digunakan.
b. Untuk pengenalan lebih lanjut mengenai *driving car simulation*, perlu adanya penambahan tentang pembelajaran akan rambu – rambu lalulintas dan penambahan animasi mobil mobil lain saat menjalankan animasi pada kondisi seperti dijalan raya.
c. Aplikasi ini menggunakan teknologi baru sehingga perlu adanya penelitian maupun pengembangan lebih lanjut mengenai teknologi tersebut. Banyak aplikasi yang memungkinkan untuk dikembangan berdasarkan teknologi tersebut. Contoh pengembangan aplikasi tersebut pada bidang medis seperti menampilkan animasi 3D pada CT-Scan / MRI, pada bidang hiburan seperti menampilkan animasi peta 3D saat pembacaan berita cuaca, pembuatan racing game, pembuatan animasi iklan 3D, dan masih banyak yang lain.

DAFTAR PUSTAKA

Ahmed, T., Hirschi, J., and Abid, F., 2009, *Flex In Action,* Manning Publication, Chicago.

Alfa Faridh, S., 2009, *Metafora Deteksi Tiupan Pada Game Flipscape",* Andi Offset, Yogyakarta.

Anonim, 2009, *Adobe Flex*, http://id.wikipedia.org/wiki/adobe-flex, diakses tanggal 12 Desember 2009.

Anonim, 2009, *Marker Detection*, http://id.wikipedia.org/wiki/marke-detection, diakses tanggal 12 Desember 2009.

Anonim, 2009, *Permainan*, http://id.wiki.org/wiki/Permainan, diakses tanggal 12 Desember 2009.

Anonim, 2009, *Rich Internet Application*, http://id.wiki.org/wiki/Rich-internet-application, diakses tanggal 12 Desember 2009.

Anonim, 2009, *Augmented Reality*, http://id.wiki.org/wiki/Augmented-reality diakses tanggal 12 Desember 2009.

Anonim, 2009, *How it's made? Blender + Papervision3D = Flash car drive* http://flashsimulations.com/2009/12/16/how-its-made-blender-papervision3d-car-drive-in-flash diakses pada tanggal 15 Maret 2010.

Haapoja, M., 2008, *Flash augmented reality – Getting Started*, http://mikkoh.com/blog/2008/12/flartoolkitflash-augmented-realitygetting-started, diakses pada tanggal 20 Februari 2010.

Haapoja, M., 2008, *Interactive Flartoolkit (Reacting To Translation And Rotation)* http://mikkoh.com/blog/2009/03/interactive-flartoolkit-transformation-from-the-transformation-matrix diakses pada tanggal 15 Maret 2010.

Jacob, S., Weggheleire D., 2008, *Foundation Flex For Developers*, Friendsof, USA.

Lott, J., Schall D., and Peters, K., 2006. *ActionScript 3.0 Cookbook*, Friendsof, USA.

Tiwari, S., Herrington, J., Elrom E., Mostafa J., 2008, *Advanced Flex 3*, Friendsof, USA.